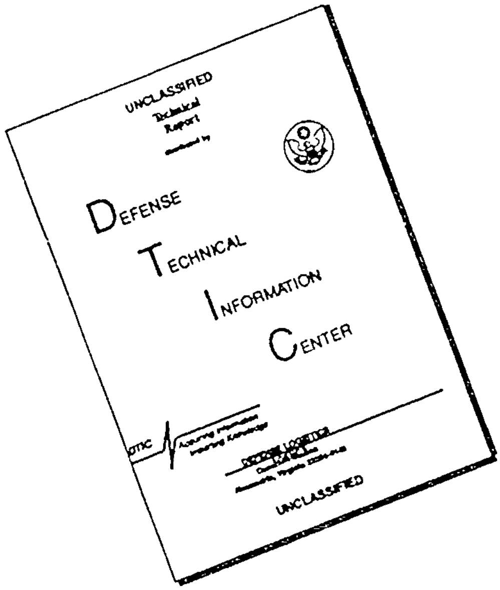
PROCEEDINGS

NAVY
USMC
AIR FORCE
ARMY
13th
INDUSTRY

INTERSERVICE/INDUSTRY TRAINING SYSTEMS CONFERENCE

AMERICAN DEFENSE PREPAREDNESS ASSOCIATION

DISCLAIMER NOTICE



THIS DOCUMENT IS BEST QUALITY AVAILABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

FOREWORD

The American Defense Preparedness Association, in cooperation with more than 160 industrial companies, Department of Defense Agencies, foreign government agencies, and the academic community is pleased to host the Thirteenth Interservice/Industry Training Systems Conference in Orlando, Florida on December 2-5, 1991.

Annually, the I/ITSC is the single most important event dealing with simulation and training technology. The Conference is endorsed by the services and establishes a forum for the exchange of information among industry, government, and academia. This year, in an effort to exploit the obvious synergism, we have combined the I/ITSC with the "Technology in Training and Education Conference" and with the "Manpower, Personnel, Training and Safety Conference". The papers published in this volume are those that will be presented by I/ITSC. The papers presented by TITE are published in a separate volume. ~~There is not a~~ separate volume for MPTS papers.

"Looking Ahead: Meeting the Global Challenge" is the theme that was chosen for the Thirteenth Interservice/Industry Training Systems Conference. This theme was chosen in the early summer of 1990, before the events of Desert Shield/Desert Storm and, the disintegration of the Soviet Union. Clearly, meeting the Global Challenge takes on greater significance in light of these recent events.

In keeping with the chosen theme, we have tried to focus the Conference on the need to sustain and improve our training technology base. The value of that training technology base was given a true test by Desert Storm and we have an opportunity to benefit from the "lessons learned" in that conflict. We can all be proud of the job that was performed by our combined forces; crushing the world's third largest military force in a matter of weeks. Training, clearly, was a major contributor to this victory and is part of the very foundation of our nation's continued strength; strength in a world in which potential threats grow more sophisticated with each passing day. By broadening our training technology base, our nation can remain dominant even in times of diminished defense budgets.

(Cont'd on next page)

The Navy is the lead service for this year's conference. As always, the personality of the conference is a reflection of the leadership of the Executive Committee Chairman. Captain Ernest L. Lewis, Commanding Officer, Naval Training Systems Center, Orlando, Florida has provided that leadership and in conjunction with the 13th I/ITSC Conference Chairman, Mr. Donald M. Campbell, has done an outstanding job of directing this year's conference. The conference brings together all levels of Users, Trainers and Educators, and Developers with their counterparts in Industry and Academia. The conference creates the environment and opportunity for the exchange of ideas and for gaining exposure to recent advances in technology, management and systems utilization.

Thanks go to the many individuals, and their supporting organizations, who have given so generously of their time and talents to make the 13th I/ITSC the best ever. They are identified on the pages that follow. Their goal has been, as in years past, to provide a forum for technical interchange, mutual understanding and individual growth within the training systems industry. The papers presented in this volume represent a distillation of the very best from hundreds of abstracts and papers that have been submitted during the past year. Your participation in the Conference assures us that our goal has been met.

Thank you for your participation; enjoy the conference.

Jack Drewett
Program Chairman

INTERSERVICE EXECUTIVE COMMITTEE

CAPTAIN ERNEST L. LEWIS, United States Navy, Commanding Officer, Naval Training Systems Center, Orlando, Florida

COLONEL B. A. SCHWANDA, United States Marine Corps, Liaison Officer/Program Manager, Marino Corps Systems, Naval Training Systems Center, Orlando, Florida

COLONEL JOHN M. GRAVOIS, United States Army, Project Manager for Training Devices, PM TRADE, Orlando, Florida

COLONEL WAYNE J. LOBBESTAEL, United States Air Force, Program Director, Training Systems System Program Office (SPO), Wright-Patterson AFB, Ohio

INTERSERVICE STEERING COMMITTEE

M. MICHAEL ZAJKOWSKI, PH.D., United States Navy, Advanced Projects Group, Naval Training Systems Center

HUBERT W. MERRY, Chief, Engineering Simulation Management Section, Ogden Air Logistics Center, Air Force Logistics Command

RALPH C. NELSON, United States Army, Chief, Program Management and Requirements, PM TRADE, Army Materiel Command

MAJOR HUNTER TOLBERT, United States Marine Corps, Project Manager, Marine Aviation Programs, Naval Training Systems Center

F. J. WINTER, JR., United States Air Force, Director, Acquisition Support, Training Systems System Program Office, Aeronautical Systems Division, Air Force Systems Command

CONFERENCE COMMITTEE

CONFERENCE CHAIRMAN, Donald M. Campbell, Daedalean, Inc.

PROGRAM CHAIRMAN, J. D. Drewett, General Dynamics Corp.

ADMINISTRATIVE ASSISTANT, Toni Vaughn, General Dynamics Corp.

CONFERENCE CHAPLAIN, Colonel Alphonse Castellana, USMC (Ret), Computer Sciences Corp.

PUBLICITY CHAIRMAN, Allen Q. Collier, Naval Training Systems Center

TECHNICAL SUBCOMMITTEE CHAIRMAN, Eric Haseltine, Hughes Aircraft Company

MANAGEMENT SUBCOMMITTEE CHAIRMAN, Judith Shellnutt-Riess, Syscon Corp.

USER SUBCOMMITTEE CHAIRMAN, Ben Blood, Coleman Research Corp.

BEST PAPER SUBCOMMITTEE, Clarence J. Chappell, Naval Training Systems Center

ACADEMIA LIAISON, Dr. A. Louis Medin, Inst. for Simulation & Training, Univ. of Central Florida

TECHNICAL SUBCOMMITTEE

CHAIRMAN

Eric Haseltine, Hughes Aircraft Company

MEMBERS

Tim Andolsek, Binghamton Simulator Company
Billy Ashworth, NASA Langley
Dorothy Baldwin, General Dynamics Corp.
Robert Barrette, CAE Electronics Ltd.
Bruce Betschart, CTA, Inc.
Dr. Arnie Bramson, General Electric
Dr. Joseph Brann, Ph.D., IBM
James Burke, Encore Computers, Inc.
Paul Chatellier, Perceptronics
Jim Chatt, Hughes Training Systems
Buddy Denham, Naval Air Test Center
Carl D. Eliason, McDonnell Douglas Training Systems
Bruce Erickson, Evans & Sutherland Computer Corp.
Ronald Ewart, WL/FIGD, Wright-Patterson AFB
Hal Geltmacher, AFHRL/OT, Williams AFB
George L. Graham, Grumman Aircraft Systems Division
John Hammond, AAI Corporation
William Holtsman, CAE-Link Corp.
Tom Hoog, ASD/ENETM, Wright-Patterson AFB
Ron Johnson, Contraves - SSI
Bruce Knerr, Army Research Institute, PM TRADE
Ed Kulakowski, Reflectone, Inc.
LCDR Michael Lilienthal, COMNAVAIR SYSCOM
Paul Little, Naval Training Systems Center
LTC Tom Mastaglio, HQ TRADOC
Dr. Bruce McDonald, Institute for Simulation & Training
Gary Nesta, CAE-Link Corp.
Gary Neuman, PM TRADE
Al Plate, ECC International Corp.
Jeffrey Pulcini, Concurrent Computer Corp.
Renata V. Schmidt, General Physics Corp.
Doug Smith, HQ US Army Materiel Command
George C. Snyder, Loral Defense Systems - Akron
Lynn Thompson, McDonnell Douglas
Stephen Trencansky, Concurrent Computer Corp.
Lois Wilson, Loral Aerospace Corp.
Judy N. Wyne, Office of Chief of Naval Education & Training

MANAGEMENT SUBCOMMITTEE

CHAIRMAN
Judith Shellnutt-Riess, Syscon Corp.

MEMBERS
Kenneth Beckham, Reflectone, Inc.
Major Bob Brier, USAF
Jim Cooksey, Palau Electronic Corp.
Dennis Duke, Naval Training Systems Center
Ron Dukes, CAE-Link Corp.
Cheval Fagan, IBM
Al Galbavy, PM TRADE
Mike Hayes, Syscon Corp.
James Horton, Loral Defense Systems - Akron
James E. Jardon II, Jardon & Howard Technologies, Inc.
L. R. "Ron" Johnson, Blackhawk Management Corp.
Carla Knapp, IBM
Major David LaBounty (USMC), Naval Training Systems Center
Frank Lawler, Syntech
Richard D. Leavitt, Computer Sciences Corp.
Jan Martincz, Hughes Training, Inc.
Steve Miller, ASD/YWS, Wright-Patterson AFB
Tom Miller, SAI Technology
Samuel B. Moore, LTV Aerospace & Defense
Ann O'Kennon, ERR Systems Corp.
Col. Gerry Patterson, HQ SAC/XRH
Cecil Ramsey, HQS Ft. McPherson
Capt. Sue Schuler, USMC, PM Training Systems
Jerry Shimp, Hughes Training, Inc.
Phillip Sprinkle, PM TRADE
Charles Sturges, Ball Systems Engineering
Frank Thweatt, PM TRADE
William J. Walker, Naval Air Systems Command
Kathryn Walsh, Consultant
Lt. Col. Raymond Wilcox, USAF Tactical Warfare Center
Sam Worrell, SBS Engineering, Inc.
Robert Zweibel, Office of Chief of Naval Operation

USER SUBCOMMITTEE

CHAIRMAN
Ben Blood, Coleman Research Corp.

MEMBERS
Lt. Col. Edward Aufderheide, NGB/XOP
Gary Baker, Boeing Aerospace
Jerry Baltz, Logicon
Lt. Col. Leroy D. Baxter, USAF
Paul Bernhardt, Daedalean Corp.
Neale Cosby, Institute for Defense Analysis
Andy Courtice, Consultant
Ben Covington, JDL, Inc.
Bernard Gajkowski, Naval Training Systems Center
Hal Hansen, US Army Training Support Center
Lou Hightower, Computer Sciences Corp.
Lt. Col. Paul Hinote, PM TRADE
Al Irwin, SAIC
Col. Wendell Kot, DAMO-TRS, The Pentagon
Len Kravitz, Reflectone
SMSgt. Gary Lewis, ASD/YWF, Wright-Patterson AFB
Jeff Marlin, Firearms Training Systems
James Marshall, Naval Training Systems Center
Pres McGee, CAE-Link Corp.
CDR Paul Miles, Office of Chief of Naval Operations
Col. Charles F. Moler, DAMO-TR, The Pentagon
Jack Nicholson, Technical Assistance, Inc.
Lt. Col. Jeff Rasmussen, MCCDC TE-35DM
Kurt Rhodehamel, Encore Computer Corp.
LTC Mark Russell, PM TRADE
Andy Seago, Office of Chief of Naval Education & Training
Ron Sheffer, IBM Corp.
Lundie Sherretz, Scientific Systems Company
Dennis Shockley, Evans & Sutherland Computer Corp.
Herb Stacy, Office of Chief of Naval Technical Training
David Tolliver, AAI Corp.
Ed Ward, Hughes Training, Inc.
Lt. Col. Frank Wells, HQ SAC/XRHN, Offutt AFB
Jim White, Training Management, Inc.
Ron Wimberley, Martin Marietta

Table of Contents

Table of Contents (cont'd)

Technical Papers	Page

SENSOR II

DISTRIBUTED INTERACTIVE SIMULATION: PRESENT AND FUTURE

Table of Contents (cont'd)

Table of Contents (cont'd)

Table of Contents (cont'd)

Table of Contents (cont'd)

Table of Contents (cont'd)

Management Papers — Page

CBT MODELS

T&E MEASURES

AUTOMATED MANAGEMENT CONCEPTS

Table of Contents (cont'd)

TABLE OF CONTENTS (cont'd)

TABLE OF CONTENTS (cont'd)

User Papers | Page

TRAINING SYSTEMS EVOLUTION

USER REQUIREMENTS

USER ASSESSMENT

EXPERIENCES IN WRITING READABLE AND UNDERSTANDABLE ADA

John Glaize, Staff Scientist
CAE-Link Corporation
Binghamton, NY

ABSTRACT

A critical and much-publicized advantage of the Ada programming language is the potential for producing more reliable, maintainable software by enhancing program readability and understandability. Many people in the programming community have wondered just how well this potential would be realized on a large-scale Ada project. Is it really easier to read and understand Ada code? The CAE-Link Corporation, utilizing the actual Ada code developed on the B-2 Aircrew Training Device, has now been afforded the opportunity to investigate this question. This paper presents some of the issues raised and the results discovered by this investigation. A critical issue is the recommended naming of language components such as packages, subprograms, parameters, types, and objects, as well as how readability is affected by the various contexts in which the components can appear. Other issues are program formatting, nesting of components, and the length and understandability of the Ada statements. The system architecture, which defines the relationship and interconnection of program components, is very important for ensuring understandability of the system as a whole. Finally, the paper addresses the training that is necessary to educate engineers in the art of writing and of reading Ada programs. The conclusion is that Ada programs are not inherently more readable and understandable, but that successful Ada development in this area requires special awareness of the issues and unique programming efforts.

INTRODUCTION

An important premise in the emerging science of software engineering is that overall quality is significantly enhanced when the software is readable and understandable. The reasons behind this premise derive from the criticality of the requirement that much of today's software, especially in leading edge applications such as Aircrew Training Devices (ATDs), exhibit the characteristics of maintainability and modifiability. Maintainability refers to the extent to which errors can be efficiently isolated and corrected, and modifiability refers to the extent to which the software can be effectively upgraded to meet new requirements. These characteristics are vital for keeping the systems operational and up-to-date, and for reducing life-cycle costs. Simply stated, the premise is: Software can be more effectively maintained and modified when it is easier for the maintainer to read and understand.

Recognizing the need for readable and understandable software, the developers of the Ada programming language designed many language constructs that support this goal. Therefore, the promise and potential of Ada is the production of higher-quality software that exhibits these characteristics. The CAE-Link Corporation, utilizing the actual Ada code developed on the very large-scale B-2 ATD, has now been afforded the opportunity to investigate the extent to which this potential can be realized, as well as some of the issues that influence its realization. Some of our initial findings, experiences, and opinions, based on this investigation, that this paper discusses are:

- The issue of software readability and understandability is very complex, multi-faceted, and subjective, depending greatly on who is doing the reading and what that person is trying to understand. Readability is increased when software writers attempt to consider the needs of several types of readers.
- Readability/understandability in Ada depends more on how the language constructs are used than on any inherent properties of the constructs themselves. Effective use often varies depending on the context in which the constructs appear.
- The use of constructs that enhance readability and understandability from one perspective may sometimes actually impede them from another point of view.
- The issue has significant impact on the nature and variety of tools required.
- Training for the engineers on the Ada language, the system architecture, the toolset, and the methodology for writing and reading Ada programs is paramount.

These findings are now explored within the context of various programming issues and software engineering topics.

VISUAL FORMATS

Link's experience indicates that the visual format of the software is one of the most significant factors affecting readability and understandability, and at the same time is one of the most subjective. Whereas the helpfulness of particular aspects of the visual presentation are generally agreed upon within our company, the effectiveness of some other aspects has resulted in long and heated debates concerning the most readable presentation, and several different opinions still prevail. This experience leads the author to conclude that the programming community will never concur completely on an "industry-standard" visual format. This paper, therefore, records some of the rationale that determined Link's choices, as documented in our Ada Programming Standard. We present this in the hope that such information may promote dialog among the various government agencies and contractors using the Ada language, and generally improve the overall quality of Ada software through such shared experiences.

While exploring various aspects of the visual presentation, our experience indicates a fairly high degree of concurrence in the areas of indentation and vertical alignment. Ada syntax contains several compound statements (such as as "if" statements, "loop" statements, and "case" statements) whose range typically extends over many lines of text. In order to understand the dynamics of a program, it is very helpful to be able to easily discern the range of such compound statements, especially when they are nested. Horizontal indentation by a prescribed number of spaces (eg., 3) of the sequence of statements within a compound statement, along with the vertical alignment of the reserved words defining the statement, is quite effective. For example.

```
If some_condition then
  perform_some_action;
elsif some_other_condition then
  perform_another_action;
else
  while a_third_condition loop
    perform_yet_another_action;
  end loop;
end if;
```

Another instance where vertical alignment proves to be especially helpful is in those areas of the program which are typically read vertically, such as declarative regions. Our experience indicates that, when looking at these regions, readers are generally interested in locating "sets" of constructs such as data types or data objects, and then discerning what various types or objects are being declared in this particular program. We have found that an effective format for presenting this information is to group all like constructs together, offset by blank lines, and to arrange the data on each line into a "tabular" format by utilizing the vertical alignment of such reserved words and special characters as "type", "subtype", "is", ":", and ":=". For example:

```
Type samples   is range 1..10;
Subtype counts is samples range 5..10;

Altitude       : meters := 10000;
Fuel_capacity  : gallons := 3500;
```

The alignment employed here aids the reader in locating the set of types and subtypes and the set of data objects, along with the names and characteristics of each individual type and data object. Some of our people feel that the extra horizontal space (such as between the words "type" and "samples" in the first line above) tends to break up the declaration. Our experience indicates, however, that this space does not, in general, unduly hamper readers when they are trying to discover all of the characteristics of the type "samples" by reading horizontally across the line.

Unlike the features of indentation and vertical alignment, the issue of capitalization, in our experience, has not enjoyed concurrence. In fact, many different opinions prevail. Some people feel that programs should be written like the examples in the Ada Language Reference Manual (ANSI/MIL-STD-1815A) with reserved words in lower case and user-defined words in upper case. Others feel that a textbook in common usage should be selected (such as Booch's Software Engineering With Ada) and the examples therein emulated with respect to capitalization. Several other permutations, such as capitalizing only the first letter of each word in identifiers or capitalizing predefined type names, have been proposed.

The Link Ada Programming Standard strongly recommends the use of lower case for most program text. The only recommended usage of upper case is for the first letter of each Ada declaration and statement, acronyms, and "special" letters such as the letter E in exponential expressions and the letters A through F in hexadecimal literals. The rationale for this has been hotly debated; however, it promotes the concept that Ada (as with other "high-level" languages) is not just a method of encoding computer instructions but is actually a programming language and therefore should be consistent with the readability precepts employed by our standard written language, English. Studies have shown that lowercase text (with letters that project above and below other letters), is easier to

read than uppercase text (with its "block-like" letter structure). Capitalizing the first letter of each statement helps to emphasize the beginning of a new statement or "new thought", just as in English the initial capital letter indicates the beginning of a new sentence.

Whereas the use of such techniques as indentation, vertical alignment, spacing, and capitalization can result in programs that are much easier to read, the incorporation of these techniques can prove to be a burden on the writer. This is especially true when modifications are being made and indentations, for example, must be readjusted. Our experience indicates that tools, such as a "pretty printer" (several of which now exist in the community), are invaluable for maintaining the desired software formats while unburdening the writer.

Another, somewhat unexpected, finding in our experience is that the effectivity of the visual format can differ depending on the medium where the program is being viewed. In our environment (and we believe that this is typical throughout the industry), software is developed, maintained, and therefore read more often on a video terminal than a printed listing. Consequently, the readability aids should be oriented more toward a terminal than a listing. An example of this is the traditional use of the page-eject to separate major program components. Whereas this is quite helpful for a printer listing, it actually impedes readability on a terminal. Link's Programming Standard calls for a comment line of all asterisks to replace the traditional page-eject as a separator. Readability issues have also indicated that video terminals that can display 132 columns are helpful.

DATA ABSTRACTION

Software engineers recognize the importance of representing the data that comprise a system by use of proper, meaningful abstractions. An important principle for defining good abstractions is to use precise and detailed specification of the data, so that the reader will understand just what the abstraction does and does not represent. Just as a mother is more likely to be successful if she tells her child, "Go to the supermarket and buy a quart of skim milk" rather than "Go get food", precise and specific data definition yields more understanding than imprecise and general definition. Ada provides many language features that aid the developer in producing understandable data abstractions. Some of these features are identifiers, typing, and packages.

Ada allows data objects to have more meaningful names by not artificially restricting the length of identifiers and by permitting the use of the underscore character as a separator. This feature results in the definition of data names that typically are more precise and therefore easier to understand, (eg., "left_outboard_engine" instead of something more cryptic like "lfouteng").

Our experiences with this feature indicate that software developers have generally recognized the advantages of meaningful names, with the result that maintainers can more easily understand what a typical data object represents, without having to refer to descriptions in supplemental documentation. Some pitfalls have been identified, however. One is the potential for names to become so long that they tend to carry "excess baggage" which actually impedes readability. For example:

```
input_output_buffer_for_refreshing_the_
  display_in_real_time_mode
```

which, although precise, is so wordy as to be unwieldy to read and understand. This problem is compounded when several long identifiers are used in a single computation, resulting in one statement which extends over multiple lines. Often, the overall meaning of the statement is obscured by having to read too much verbiage.

Another, more subjective, pitfall is that some developers feel that understandability is compromised when familiar, time-honored abbreviations and acronyms are replaced by spelled-out words (eg., is "greenwich_mean_time" really more understandable to experienced engineers than "gmt")? These engineers feel that the extra effort expended in reading more words make it difficult to determine the nature of the calculations being performed on the data and actually impede the understanding usually derived from seeing the more familiar names.

Link manages the use of identifiers in several ways. The Ada Programming Standard provides guidance on proper length (eg., identifier length shall not exceed 40 characters or 7 words). The Standard is based on human factors, such as the amount of information that can be retained as a single thought. A list of approved abbreviations and acronyms, based on recognizability, has been compiled for use in formulating identifier names. Abbreviations and acronyms not on the list are discouraged and are only to be used when accompanied by explanatory comments. Finally, all code is reviewed by peers and potential maintainers during a "code walkthrough" to judge the reasonableness and understandability of the identifier names from the point of view of the typical reader.

Another Ada feature supporting data abstraction is the ability to define classes of data called "types". This allows more precise data definition by affording the developer the ability to make assertions about the nature of the data. This information is then available to the reader which can greatly aid understanding. One form of the Ada type feature is to define scalar types which represent a single quantity. An example of a scalar type and a data object of that type is provided by the following statements:

```
Type radians digits 7;
Radar_sweep_angle : radians range 0.0 ..
  3.14159;
```

These statements impart to the reader the information that the quantity radar_sweep_angle is not only a floating point number, but that it also consists of at least 7 significant digits, is expressed in radians, and assumes values of 0 to 3.14159 (Pi). Such information is very valuable if the reader is to understand the nature of the data.

Our experience shows that a potential drawback of having separate types for each class of numeric data is that Ada then requires "type conversions" or "overloaded functions" for combining different types of data in equations. Our application requires this combination quite often (eg., force = mass * acceleration). The "overloaded function" solution is often detrimental to our real-time requirements, and the "type conversions" were found to impede readability due to the addition of type names within the equations. Our solution to this dilemma is to define numeric types as subtypes of a few "base types". The subtype definition allows the characteristics of the data to be stated precisely without requiring type conversions.

Even more powerful abstractions can be defined using Ada's "composite types", especially records. In this case, all of the attributes (or component data) that comprise a given abstraction can be collected together. For example:

```
Type fuel_tanks is record
   Capacity      : gallons;
   Quantity      : gallons;
   Temperature   :degrees_celsius;
   Pressure      : pounds_per_square_inch;
end record;

Left_outboard_tank : fuel_tanks;
```

Our experience in this area indicates that it is often helpful to name a type with a plural word such as "fuel_tanks" since the type represents a class of entities. Also, the names of the record components should remain concise since they always appear within the context of the overall record name. For example, even though the component name "fuel_tank_capacity" is more precise when viewed alone, the name "capacity" is actually better because it is referenced in the context of the record name "left_outboard_tank.capacity" which reads more logically than "left_outboard_tank.fuel_tank_capacity".

Ada carries the concept of data abstraction to a dimension not possible in some earlier languages by providing the package construct. This construct allows the program developer to "package together" not only a data type but also the operations that apply to that data to form a very powerful abstraction. The system architecture that Link has employed on the B-2 uses this capability to significant advantage.

An interesting fallout of the package construct relating to readability issues is that the names of entities within the package occur within the context of the package name. As we saw above, the context of record components should be considered; likewise the context of package components should be considered. For example, if the package name is fuel_tank and the data type contained within the package is fuel_tanks, the result can be a fairly strange looking construct such as:

```
Left_outboard_tank : fuel_tank.fuel_tanks;
```

Even though it may first appear too cryptic, we find that a better name for the data record might really be something like just "data", because, when taken within the context of the package in which it appears, the result is the complete name:

```
Left_outboard_tank : fuel_tank.data;
```

Our experience shows that Ada's data abstraction features, containing such constructs as packages and records, require a bit more care on the part of the software developer so that the full "dot-notated" name is readable.

PROCEDURAL ABSTRACTION

Ada extends the procedural abstraction capabilities of some other languages by allowing more meaningful names for procedures, functions, and their associated parameters. Our experiences in this area indicate that Ada procedures represent an abstraction of an action and are therefore more readable when named by a verb or verb phrase. Ada functions, on the other hand, are an abstraction of the value which they return, and therefore should be named by a noun. The names of the formal parameters result in enhanced understandability when they indicate the nature and use of the parameters. An example of a procedure call that illustrates these principles is:

```
Determine_the_display_window_limits
   (producing_the_upper_limit    =>  left_window_upper_limit,
    and_the_lower_limit          =>  left_window_lower_limit,
    using_the_cursor_x_setting   =>  left_window_cursor_x,
    and_the_cursor_y_setting     =>  left_window_cursor_y);
```

This format affords the reader a great deal of information, especially when the entire statement is read from beginning to end like an English sentence. It is clear what action is being taken, what data is being produced by the action, and what data is used to perform it. Ada even allows the writer to exploit "connecting" words (such as "using" and "and" in the example above) to tie the entire abstraction together.

This example also demonstrates our experience with the varied aspects of the readability issue. Readers have several objectives in mind when reading software. In a normal English sentence, the embedded spaces in the statement would not appear because these spaces tend to interfere with understanding the thought expressed by the sentence as a whole. However, our experience shows that when reading a piece of software, some readers are more interested in the use of particular parameters rather than the procedural abstraction as a whole, and the embedded spaces help their eyes to focus on these parameters. Our experience also indicates that leaving the spaces in the code constitutes a reasonable readability compromise: the spaces are an aid to those readers who wish to concentrate on particular parameters and do not appear to present undue distraction to the readers of the entire statement.

Another of our findings that is illustrated by the above example is that readability factors can often vary depending on the context in which they appear. The example shows that some of the very features that enhance readability from the point-of-view of the procedure call statement can result in stilted and abnormal constructs when viewed from within the implementation of the procedure. In this example, the situation occurs with the names of the formal parameters. The formal parameter names contain connecting words that help the flow of the code in the call statement but can lead to strange-sounding statements within the procedure, such as:

```
Producing_the_upper_limit :=
   and_the_cursor_y_setting + 128;
```

Here the words "producing" and "and" are meaningless outside of the context of the call statement, actually, they are worse than meaningless because they make the code inside of the procedure read in a silly, unnatural way. We found that a reasonable way to manage this situation is to provide an "internal" name within the procedure when the "external" name causes trouble. This is accomplished by using Ada's "renames" feature. For example, the statements:

```
The_upper_limit       : ... renames
   producing_the_upper_limit;
The_cursor_y_setting  : ... renames
   and_the_cursor_y_setting;
```

appearing in the declarative region of the procedure lead to the more readable statement:

```
The_upper_limit := the_cursor_y_setting + 128;
```

COMPONENT UNDERSTANDING VERSUS SYSTEM UNDERSTANDING

One of the more intriguing and unexpected of our experiences is the apparent dichotomy between the factors affecting the understandability of the individual software components versus the understandability of the system as a whole. Indeed, we are often plagued and bedeviled by the situation that certain constructs which result in increased understanding at the component level actually can prove to be a detriment to the understanding of the workings and interconnections of the software when viewed as an entire system.

One example of this situation is the dilemma of determining an effective utilization of Ada's "use" clause versus "expanded names". Consider the following statement:

```
X_coordinate :=
mathematical_operations.cosine (apex_angle)
   * radius;
```

In this case, the "cosine" function is included in a package named "mathematical_operations" which is required as part of the expanded name if the "use" clause is not included.

From the standpoint of understanding the logic of the component in which this statement appears, most of our engineers feel that it is unnecessary to know where the cosine function is defined, and that this longer name actually impedes the understanding of the mathematical significance of the statement. This is felt to be especially onerous when

the "additional words" of the expanded name(s) cause the statement to extend over several lines. Also, the names tend to "overpower" the single character mathematical operations (such as the "*") with the result that the way in which the various terms are combined is obscured by the effort required to read the long names of the terms themselves. In mathematical operations both the terms and the operators must be understood.

The names and the resulting statements can be shortened by utilizing the "use" clause. Since this makes the mathematical functions directly visible, the above statement can now be written:

```
X_coordinate := cosine (apex_angle) * radius;
```

which most of our engineers feel to be easier to read and understand. The difficulty with this construct occurs when engineers want to understand the system beyond this particular component (for example, if they suspect that the cosine is returning an incorrect or incompatible value and wish to inspect its specification or implementation). It is now less obvious where this function is located. The problem is compounded on a large system such as the B-2 ATD which is comprised of thousands of components.

Our experience shows that a reasonable compromise to this situation is to utilize the Ada "renames" clause to accomplish a feat similar to the internal and external names for procedure parameters described earlier. In this case, the "renames" capability allows expanded, more explanatory names for entities outside of the component to be replaced by shorter, "internal" names for use within the implementation of the component. An example, which would appear at the beginning of the component, is:

```
Function cosine ( ... ) return ...
   renames mathematical_operations.cosine;
```

The reader is now able to determine the package in which the cosine function is defined without the increased encumbrance of this information being present at each invocation of the function.

A second example in our experience of the apparent conflict between component understandability and system understandability is attributable in part to the real-time architecture that we employ on the B-2. As stated earlier, our architecture is strongly based on an object-oriented paradigm which encourages the production of cohesive, self contained, and potentially-reusable components to implement the necessary abstractions. An attendant characteristic of such components is that their design should not depend on the source of their inputs nor the destination of their outputs. This characteristic increases the potential for taking the components to another application and "plugging them in", because they should work as long as the interface remains consistent.

The concept of object orientation greatly enhances the understandability of individual components from an internal point-of-view. All of the relevant aspects of the abstraction are encapsulated within the software component that implements that abstraction, and therefore, the internal workings of the abstraction can be completely understood without requiring the reader to have knowledge of the entire system. Our experience shows, however, that (especially during system integration checkout), many engineers have to understand the software on a more global basis in order to determine and verify the role of individual components within the context of the particular application. The very architecture that enhances understandability of the individual pieces can sometimes impede understandability of the system. The interfaces and system data flow are contained in other packages and can be more difficult to locate and fathom.

Link feels that the advantages of the object oriented paradigm are worth retaining, and therefore, augments the Ada software structures with a tool called the Interface Management Data Base (IMDB) to help engineers identify and determine the overall interface and data flow mechanisms that knit the components together to form the entire system. The IMDB contains information that defines the source and destination of all interfaces for a given ATD, along with the capability of producing reports to help engineers gain understanding of the way in which the entire device is assembled from the various components.

A third example of the conflict between some various "levels" of understanding that we have experienced has a slightly different slant than the first two examples. In this case, the design decision involved is whether, from an understandability point-of-view, a package containing several subprograms is better implemented as a single physical package or as a set of subunits with the parent package containing "is separate" clauses.

This decision appears, like so many others, to be subjective. One group of our engineers feels that implementing the subprograms as separate subunits provides the reader of the package with a convenient "list" of the various procedural abstractions that are contained within the package. The reader does not have to "wade through" the code that implements one subprogram in order to locate the

next subprogram. For example, a package body with the following layout:

```
Package body fuel_tanks is
   Procedure compute_quantity ( ...) is
     separate;
   Procedure set_demanded_quantity ( ...) is
     separate;
   Function temperature ( ... ) return pounds
     is separate;
end fuel_tanks;
```

allows the reader to easily see what abstractions are present.

Another group of engineers feel, however, that this structure impedes understandability due to the inconvenient logistics of switching between different compilation units (and probably different disk files) when trying to understand the internal implementations of the various subprograms and how they interact with one another. They prefer to see the code for all of the subprograms contained physically within the package body compilation unit.

As in the earlier examples, the apparent conflict between these two opinions seems to depend on what level of understanding the reader is seeking to attain. When attempting to understand the package as a whole, the first structure appears to provide the advantage; when attempting to understand the internal workings of the various package components, the second may be preferable. The Link Programming Standard does not state a "hard-and-fast" rule on this subject, because neither view has been found to have a clear advantage. Our experience indicates, however, that if subunits are used, the software development environment should contain a tool that helps engineers locate the files in which the subunits reside (eg., Digital's Source Code Analyzer).

TRAINING

Our experience shows that training is even more crucial when using "new techniques" such as the Ada language and object-oriented design than it used to be in the past. Understanding of Ada programs is much easier when engineers understand not only the Ada syntax, but also the rationale behind such constructs as packages, and how they support software engineering principles like data abstraction and information hiding. It has also proved valuable to train engineers in the features of the system architecture so that they can more easily discern the use of the various program components, and how these components are assembled to form the complete system. The look and feel of an Ada program takes some "getting used to". Engineers who are reading or writing Ada software for the first time experience less trouble and trauma when they are given the advantage of some training on how to write Ada and on how to read Ada.

CONCLUSION

We believe that our experience indicates that the potential for higher-quality, more readable and understandable software can be realized using Ada. The language is only the first step, however, and must be augmented by procedures and practices, standards, training, tools, and the judicious application and dissemination of lessons learned.

ABOUT THE AUTHOR

John Glaize is a Staff Scientist and Ada Specialist at CAE-Link Corporation, where he has held various management and technical positions over the last 18 years. His primary task for the last five years has been to define a real-time Ada architecture and software development methodology to maximize the production of maintainable and reusable software. He holds a B.S. degree in mathematics from Carnegie-Mellon University and an M.B.A. from SUNY-Binghamton.

Ada Types:
The Cornerstone of Simulation Models

By David C. Gross and Lynn D. Stuckey, Jr.

Boeing Defense and Space Group
Missiles and Space Division
Huntsville, Alabama

ABSTRACT

System simulation is the definition, control, and implementation of algorithmic models that replicate a system's real world behavior. Developing a useful simulation model requires a clear abstraction of the system. Software engineering supports abstraction by imposing a consistent structure on objects. One structural feature introduced by recent programming languages is strong [data] typing, aiming at two benefits: clarification of the design and enhancement of model verification. Strong typing clarifies the design by controlling the characteristics of an object, and enhances model verification by revealing errors early in the design cycle. Designers have traditionally viewed strong typing only as over-restricting the mixture of data units (e.g., meter versus degrees), an experience which has left a bad taste in many mouths. However, strong typing is a multifaceted tool which can apply to a broad range of software design problems. Simulation model designers can use Ada types to define, control, and implement models yielding:

(1) requirements consistency and traceability,
(2) interface definition/control,
(3) maintainability,
(4) reusability, and
(5) portability.

Because designers imagine and implement complex systems in parallel, projects can suffer from the fracturing effect of multiple visions of the final product. Strong typing can unify the system design, however, strong typing is only a tool -- the availability of which does not ensure its correct application. The challenge is to successfully implement it. This paper examines the successful use of Ada types for the design of simulation models, and points out the pitfalls of extreme approaches such as no-typing and over-typing. It presents Ada types as a scheme for enforcing a single system structure and as a foundation for generic simulation models. Finally, the paper discusses how types impact the software's lifecycle.

BACKGROUND

It comes as no surprise that designers implement the majority of a training device's *simulation* (as opposed to emulation) in software. Software is flexible, adaptable, and can simulate systems which have never existed. On the other hand, software has traditionally been a source of introduced errors and frustration for the system designers. The increased use of software for training simulation has pushed the state of the art in systems simulation.

System simulation is the discipline which attempts to construct useful models of real-world systems. The term "model" evokes images of stick and clay figures, or perhaps the plastic airplane models that children glue together; and this is precisely what we mean. A model is a representation of the system of interest, instantiated in a different medium and with insignificant differences from the "real" system. We construct the model to study some relevant aspect of the system such as its appearance, size, speed, range, mass, etc. Obviously, the model must be cheaper and easier to build than the real system, or we would use the real thing. The model should give us the ability to try experiments we cannot try with the real system, limiting our risk and expense.

It is a large intuitive leap from plastic toys to a multi-million line (and no doubt, multi-million dollar) software model of a weapon system. Nevertheless, the principles involved are the same. The primary difference is that the "map" from the model to the real system is substantially more esoteric than the map from a toy car to the family car. Building a software model creates a tension between the real system's physics and the simulation software's syntax; *both* of which exist only as images in the developer's mind. Modeling is a very creative act. The process by which a simulation maps an aircraft in flight to a software model, and then to ones and zeroes, has traditionally been the domain of arcane specialists. Particularly intimidating to the eventual user, this process generally requires three groups of "experts" (e.g., software, aerodynamic, and systems engineers) who do not even understand each other! This is not the kind of well-defined, disciplined process that inspires confidence in the resulting product.

The clarity of the model's map to the real system is the basic factor for determining the software's understandability, and in effect our confidence in the model. This "map" is formally called the simulation's system abstraction. Our confidence in the simulation is driven by how well we grasp and comprehend this abstraction. Therefore, the critical issues in system simulation development involve developing abstractions. How do we create an abstraction? How can we control the abstraction while the code is evolving? How can we build an abstraction that supports shifting requirements? How can we build an abstraction that communicates well with other parts of our team? Successful answers to these questions will result in production of high quality simulation models.

Fortuitously, the software engineering discipline, aimed at defining and refining the software process, has risen to prominence as a system simulation tool. In fact, one of the primary concerns of software engineering is applying abstraction to software projects. One important development of the software engineering effort is the Ada programming language. The Department of Defense sponsored the development of Ada specifically to address the goals of software engineering (i.e., modifiability, efficiency, understandability, and reliability). They intended for Ada to reach these goals by employing software engineering techniques such as abstraction, information hiding, modularity, uniformity, completeness, and confirmability. A brief investigation of the language will reveal that the rich "typing" feature is a primary means by which the language designers intended to include these qualities. Of course, it is by no means clear that they succeeded, or that the Ada types by themselves are sufficient for high quality system simulation. The following sections present a detailed discussion of modeling techniques as implemented in Ada via types. We begin with a brief introduction to Ada types. We follow with a discussion of software modeling. Finally, we present a look at implementing modeling via Ada types.

ADA TYPING

Although this is not a tutorial on syntax or semantics of Ada types, a brief digression into the nature of typing in the Ada language is in order. Almost every language (including assembly languages) provides for some kind of data typing. Typing provides the programmer with the capability of implicitly classifying data. All data is represented in the computer by an underlying set of bits, but the type of the data provides an implicit definition for interpreting the information. For example, the bit pattern "01000001" could be interpreted as the integer 65 or the ASCII character "A". FORTRAN 77 supports a fairly typical set of types including integer, real, logical, complex, and character, along with arrays of any of these types. The programmer "declares" that he desires a variable of some type, and generally can specify the amount of memory allocated to that variable (bit, byte, word, etc).

Ada provides for the fundamental types mentioned above, as well as extending the concept. Although other languages introduced many of these concepts, Ada collected all of them and expanded their power in a language intended for common use. Most of the additional type features provide the capability to extend the inherent types available in the language. The exception is that Ada provides two real types, fixed point and floating point. Figure 1 illustrates the structure of Ada types.

The extension to typing with Ada provides powerful and expressive capabilities to the modeler. Ada controls access to data, even while it is being used, through of the "private" (and limited private) type extension. The private type permits procedures to perform limited operations on a data structure without gaining read/write access to the data. Ada also extends the concept of typing to intertask communication. The task type permits a Ada task to create and communicate with executing entities in the computer sys-

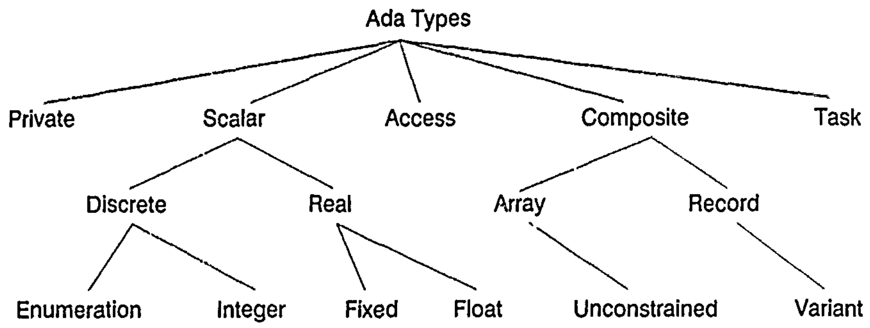


Figure 1

tem. Ada includes an access type which gives the programmer direct access to memory locations. Ada provides a subtle extension to the array type providing for unconstrained arrays, permitting the programmer to design an operation on an array of unknown length. The extension for record types permits the free association of data from mixed sources and types into a single "envelope" for manipulation. Variant records provide the capability to build a single record structure which can be tailored at declaration to support different, though similar entities (e.g., airplanes and trucks). The most pervasive extension beyond the fundamental types is the "enumeration" type. In an enumeration type, the programmer can specify the exact values permitted a variable of the new type. Figure 2 illustrates a new enumeration type declaration and declaration of two variables of that type.

```
type Colors is (Red, Yellow, Blue, Black, Grey);

Sky_Tone : Colors;
Sun_Tone : Colors range Red..Yellow;
```

Figure 2

Ada provides a set of operations for types that improve the programmer's ability to manipulate the data. Ada provides derived types to distinguish between data with similar appearance which should not mix (i.e., distance and temperature). Ada provides subtypes to further constrain an existing type (i.e., summer months from months in the year) while permitting ready conversion. The most powerful typing feature Ada provides is the rich set of type attributes. The language defines attributes over every discrete type such as 'first, 'last, 'succ, and 'pred which provide visibility into the type without coding the explicit value. Further, the special 'image and 'value attributes automatically translate between discrete type values and text representations. Ada provides every type with attributes such as 'address, 'size, 'constrained, and so forth. One might worry about converting between all these types, but Ada provides for casting between scalar types for conversion as well as unchecked conversion for any type. Finally, the programmer can specify the bit pattern for any type's representation, which provides access to the raw machine.

MODELING

A simulation model is a system, that is, a set of interrelated entities working together to achieve a common purpose. Obviously, the purpose of a simulation model is to mimic the behavior of the real system within a set of characteristics. These significant characteristics are the "state" variables of the simulation, because their value at any given point describes the state or condition of the system -- in so far as the simulation user cares.

The decomposition methodology employed for the model defines the set of model entities. In a classic functional decomposition, these are the functions of the real system: flight dynamics, weapons, electronic warfare, etc. In an object-oriented approach, the entities are the major object classes of the real system: terrain, culture, platforms, projectiles, etc. The capabilities of the user must drive the selection of methodologies. In our experience, we have found that most users relate well to an entity breakdown by objects, because they understand the notion of "laying my hand on it". Notice that this is really the issue of "how clear is your map" revisited.

Simulation software defines the interrelationships between entities by the data structures used to communicate between them. There are three general approaches to this problem: (1) common global data, (2) parameter lists, and (3) message passing. To build well-accepted simulations, we

must levy constraints on these data structures. There are three desirable constraints for the data structures which define the interrelation between entities: (1) access, (2) quantitative, and (3) qualitative. The model must prevent access to data by entities which should not have it (e.g. threats should not know the "real" position of the target). The model must control the allowable ranges for data values (e.g., prevent the switch from exceeding the number of positions). Finally, the model must restrict time-oriented changes in the data (e.g., platforms should progress smoothly through space).

Software System Engineering

If the simulation model is a system, then it has a definite life cycle, with full development, production, and operation phases. Figure 3 illustrates the lifecycle for any system. The central motivation for adopting a systems engineering approach to design is that almost all designs overemphasize the development phase at the expense of production and operation. This is just as true of software designs. Model designs must support trainer integration (production). We need to develop models that support changes concurrently with the real system (operation).

sume real-time computational resources; but implicit defensive practices are largely a function of compile time checks. We can go even further by removing the implicit run-time checks after the code is throughly tested, by setting a single compiler option.

Ada types provide support for specific modeling problems. In most languages, we model discrete states by assigning integers for each state; 1 for orbit, 2 for flyout, 3 for rendezvous, 4 for refueling, etc. The corresponding Ada type would be "type Tanker_State is (Orbit, Flyout, Rendezvous, Refueling);". Ada allows us to make this map explicit and to restrict the variables that contain the state to the appropriate values. Furthermore, Ada provides a way (via type modifiers) to explicitly state the range, interval size, and digits of precision for types, and variables within a type. This has the potential to eliminate, with a single declaration, thousands of "IF" statements scattered throughout the simulation which intend to protect data, each coded slightly differently. Through subtypes and derived types, Ada allows the programmer to directly model subtle differences between data. Via unconstrained types Ada allows the programmer to delay the decision of the final data size until it is available

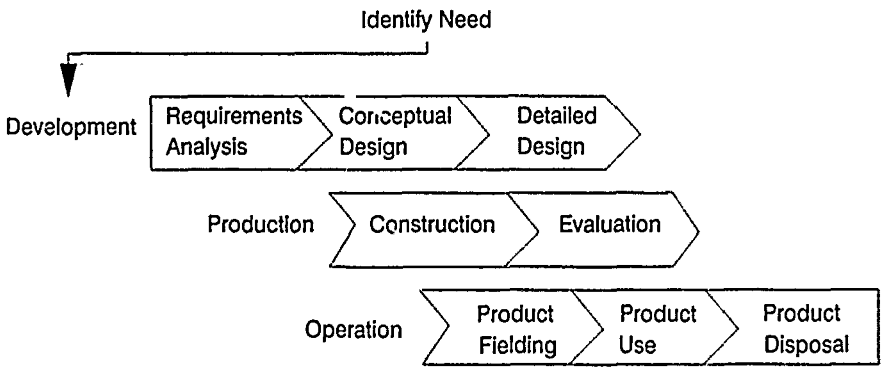


Figure 3

How do Ada types support model development? The answer is that they shift the defensive programming burden from the error-prone programmer to the compiler. The compiler has the advantage of repeatability -- it yields the same answer every time. Smart programmers code to protect their interests; checking for list overflows, illegal values, nonsense equations. But in Ada, this *explicit* defensive programming is shifted from the back of the programmer to *implicit* defensive programming in types enforced by the compiler. Real-time applications gain an additional benefit. Explicit defensive practices con-

-- if it ever is. Again, this eliminates the need for explicit defensive programming in thousands of dispersed locations, each coded ever so slightly different. Finally, access types provide for data structures which dynamically grow and shrink according to need during the simulation. This stops the practice of always claiming the maximum memory and maintaining a pointer to the "last" used spot.

How do Ada types support model production? Software production is mostly an integration effort of code developed by widely dispersed

groups. Ada supports integration *in development* through types. Correctly designed Ada projects use a tightly coupled and controlled set of types packages. These "project type" packages define the initial baseline for the software. Every compilation must use the baseline set of types. Every time the developer recompiles his code, he must resolve his changes against the baseline. Essentially, these packages provide a virtual representation for the rest of the software.

Of course, the specific modeling benefits derived in model development spill over to help the rest of the software life cycle. The major production advantage is that the types have become an extension to Ada, precisely tailored to support the current project. Much of the exhaustive hand checking of data structures is replaced by completing a successful compilation. Our experience is that integration times for Ada systems have been sliced by an order of magnitude over comparable FORTRAN projects.

How do Ada types support model operation? Operation in the context of software systems involve the discovery of delivered bugs and capability upgrades. As mentioned, the user can turn type checking on or off to support problem investigation and solution testing. Our analysis has shown that more than 80% of Ada types in actual use are not fundamental types (real, integer, etc.). In fact most models depend heavily on the use of enumeration types. Since the algorithms for manipulation of these data structures need not depend on explicit knowledge of the underlying types, many capability changes simply involve changing the options in an enumeration list, and recompiling.

Design Clarity

The most difficult contrast to understand between hardware and software engineering is that for software, the design "is" the product. The implication of this is that the design's clarity is much more a driver of the product's quality than is the case for hardware. Software engineers sometimes state this idea as "the software is read many more times than it is written".

Given this, we can begin to see how important strong typing can be to improving programmer productivity. Which design is clearer, a set of variables all of type "REAL", or the same set with types "Temperature_In_Centigrade", "Airspeed_In_Knots", and "Power_In_Watts"? The multiple type set is clearer because it presents a higher fidelity map from the real system to the simulation; it reduces the mental burden to understand the design. Obviously, it is easier for a new programmer to understand the high fidelity map, and thereby make real contributions to the effort. Likewise, the high fidelity map is easier for the user to understand, increasing the chance that he will believe in the product.

Model Verification

Model verification is concerned with the effort to establish that the software is operational. The most simplistic verification is compiling the code. Clearly, the effectiveness of this verification is determined by the throughness of the compiler. The typical compilation only investigates the code for syntax errors. On the other hand, types empower the Ada compiler to investigate the semantics of the code. The Ada compiler will inspect every procedure call and every equation for types correlation. Perhaps the best part is that the programmer determines the degree of detail in this investigation by his choice of types (i.e., derived versus subtypes).

A number of more involved techniques are available for verification: modular design, peer review, traces, sample runs, animation, and data analysis. Private and generic types support a modular design by giving the code only the access needed to do the job. Ada types support peer reviews because the code is more expressive of the model -- the names and options for types generally are defined by the model's requirements document. The remaining techniques are supported by the software development environment.

Model Validity

Model validation is concerned with the effort to establish that the model accurately represents the real system. The critical validation criteria is that decisions based on the simulation should be the same as decisions based on the real system (if available). There is no such thing as an absolutely valid model, therefore we can validate it only for a given set of conditions. For example, a cockpit procedures trainer is not appropriate for combat training.

Given the expressive power of types, we can develop a model with a high face validity. The names of types, the options within types, the data ranges, and so forth come verbatim out of the model requirements document. The strongest advantage is that the entire data requirement is captured in a single types declaration. The reviewer is not forced to track down every use of a piece of data to see that the local code that

enforces the constraints on the information. This has the helpful effect of reducing the model's complexity for the validation process.

Model Credibility

A credible model is one accepted and used by the customer. Credibility is often overlooked by the developer -- he simply assumes the user will love it! On the other hand, users will occasionally take a non-verified and non-validated model as credible. Developers (who plan to stay in business) must strive to make their models credible. Models which conform to expert opinions, observations, and existing theory about the system tend to be highly credible. However, no one is going to take your word for it that your model possesses these qualities. The user desires to see for himself, and see it in your code. Once again, the expressive power of Ada types directly impacts the ability of non-experts to accept the model, increasing the model's credibility. The very syntax of types contribute to the model's credibility because their language and phrasing can exactly match the real system's self-description. Finally, the model with Ada types gains credibility from the discovery of many errors at compilation, meaning that fewer errors survive to be seen by the user.

IMPLEMENTATION

Implementation is the translation of a model from concept to reality. Types are the primary vehicle for implementing the model in Ada. Careful use of types results in software with desirable qualities such as requirements traceability, interface control, maintainability, reusability, and portability. We know these are desirable qualities because they result from applying the goals and techniques of software engineering. Ada types can be the tools we use to achieve the goals of software engineering.

Requirements Consistency and Traceability

One of the most crucial parts of model design is the ability to show consistent and traceable requirements. The only way to make the assessment of whether the simulation is a sufficient model is to be able to view and test the design requirements in the simulation itself. In the past, requirements were often overlooked or lost in the heat of the code and integration phases. Until the requirements become an integral part of the code, the implementation will diverge from the requirements. This divergence has historically been a problem because the resulting design fails to achieve its common goal: a system which fulfills the set of design requirements. Simulations that fulfill their entire design requirements are rare and simulations that have any direct traceability to these requirements in the code are rarer still. And lest we forget, traditional after-the-fact documentation is *not* requirements traceability. Traditional documentation is only dreams of what should have been in the code. True documentation is based exclusively on the code (not the comments). True documentation reveals the actual requirements implemented in the code.

Design requirements are a broad expression of what should and should not be done in a system. Types are the vehicle to express these requirements in code. The single greatest advantage of enforcing requirements through types is that we create compilable requirements. This gives us an *objective* test as to whether the design has implemented the requirements. This forces a system mindset from the beginning of the program. Compilable requirements require extended effort in analysis and design. But, they reduce problems and inefficiencies later in the program. By using types for requirements consistency and traceability, we are able to promote uniformity and confirmability in the simulation. Types can be a readable description of the system. Types can document the source of driving requirements. Types can also restrict object interaction in a way that is reflective of the real world. Types can police the simulation and training constraints. A types package can function as a central location for all system unique features.

Ada provides a number of typing features to encapsulate the requirements analysis in compilable code of which the most important is the enumeration type. Enumeration types provide the ability to really express requirements in code instead of hiding them with "magic numbers". Other types that are of use are (1) subtypes to restrict ranges without restricting interaction, (2) derived types to restrict interaction, (3) private types to restrict visibility, and (4) generic types to share algorithms among different instances of objects. The effect of all of this is to provide automatic universal data constraints, precisely as required, and with a large degree of visibility.

In general, requirements arise from three sources: intrinsic requirements of the implementation, design criteria, and system analysis. Requirements intrinsic to the implementation are captured by any language -- they are only required because

of it! The requirements traceability we have in mind addresses those arising from the design criteria and system analysis. The expressive power of Ada types provides a way to capture these requirements directly in the code in the natural language for the problem. A code example of the use of enumeration types to express a requirement rising from design criteria is found in Figure 4.

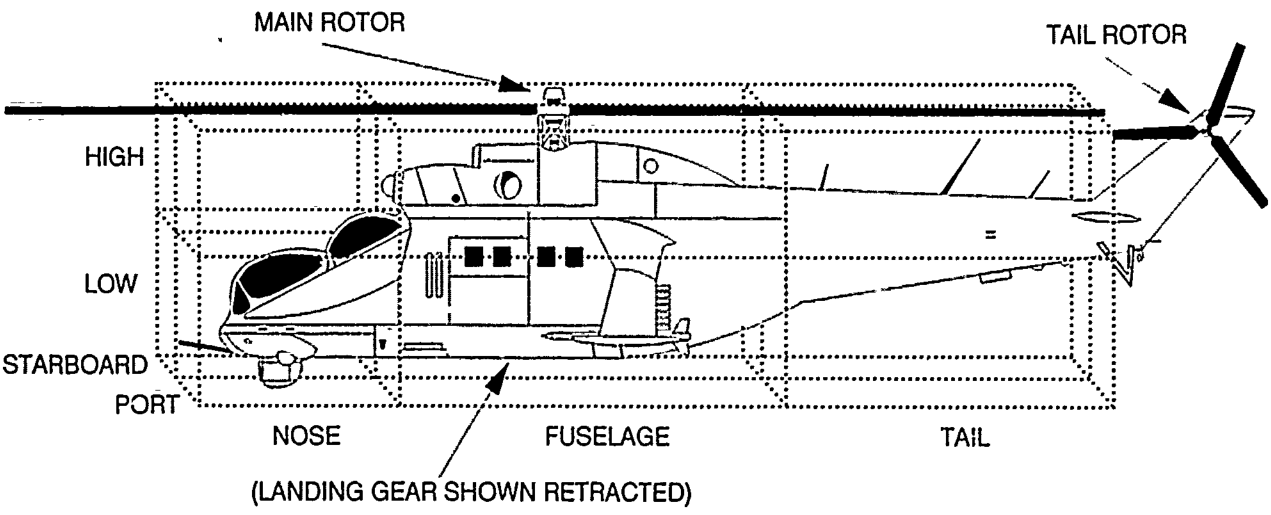
```
-- DOT Contract FA-75WA-3650
-- Programmable Test on Wind Shear

type FAA_Approved_Wind_Shear_Profiles is
  (Neutral_Logarithmic, Frontal_2_Logan,
  Thunderstorm_2_Philadelphia, Thunderstrom_3,
  Thunderstorm_4, Thunderstorm_5,
  Thunderstorm_6_JFK, Frontal_3,
  Thunderstorm_FAA_Mathematical);
```

Figure 4

What about a design requirement arising from system analysis? Figure 5 shows a drawing from a system specification and the resulting type declaration. The requirement states the need to determine the detonation of a weapon on the aircraft classified by one of fifteen zones on the body. This is to alert the affected systems of the need to consider the possibility of damage to their systems. The illustrated type implements this directly. A separate requirement to state the severity of the damage would be implemented by wrapping this type into a record with an additional field for severity.

Interface Definition/Control

In the real world, a system's interfaces are of paramount importance. How the subsystems plug, wire, bolt, or connect together is most of the design problem. With this in mind, it is astonishing that system simulation has done such a poor job of modeling interfacing. Part of the problem is that the simulation's decomposition has not represented the real world system. Designs based on functional decomposition do not lend themselves to comparison with the real world interfaces. The increasing prominence of an object-orientation is bringing interface modeling into focus. Another problem with interfacing has been the water bucket approach; throw all the interfaces into a bucket which anyone can access. This common global data approach has been a large source of error in most simulations that use them. For a simulation to become functional without extensive wasted effort, the interface within the simulation must be both defined and controlled. The simulation software itself must directly support true interface definition/control.




```
type Damage_Location is (Main_Rotor,          Tail_Rotor,          Landing_Gear,
  Low_Nose_Port,      Low_Nose_Starboard,      High_Nose_Port,      High_Nose_Starboard,
  Low_Fuselage_Port,  Low_Fuselage_Starboard,  High_Fuselage_Port,  High_Fuselage_Starboard,
  Low_Tail_Port,      Low_Tail_Starboard,      High_Tail_Port,      High_Tail_Starboard);
```

Figure 5

Ada types provide for interface definition and control via compilable interfaces. This directly parallels the real world practice of interface definition. Any type can be an interface, but records are especially useful. Record types are a collection of basic types, directly modeling the interrelationship of diverse data concerned with a common subject. This collection captures in code the interface between system components. These interface types form the parameter lists of subprogram specifications. When compiled, these specifications form an interface *contract* between the subprogram and its user. The interface types result from a dataflow analysis and become the compilable program design language (PDL). Ada types provide direct support for designing the interface model.

Another concept in interface definition/control is the understandability of the interface. When we look at an electrical plug, we understand the types of its interfaces, i.e., neutral, ground, and hot. We also understand to what it would interface. The same must be true of simulation software. Via Ada types, we can describe the interface, its units, its purpose, or even its origin. Notice that use of interface types in software supports the direct production of the interface definition document (IDD). The IDD derives from the code -- not some extraneous piece of documentation. An example of a compilable interface is shown in Figure 6.

```
type Moving_Model_State is record
  Position    : Gaming_Area_Position_Components;
  Orientation : Angular_Position_Components;
  Velocity    : Gaming_Area_Velocity_Components;
  Rotation    : Angular_Velocity_COmponents;
end record;
.
.
.
procedure Update_Position (
  Model : in out Moving_Model_State);
```

Figure 6

Maintainability

Maintainability is the degree of difficulty in continuing to use the software in the face of changing equipment, requirements, and personnel over the life of the project We desire maintainability because the operational cost of software can be significantly greater than the development cost. If a simulation is not maintainable, the cost of changes will be excessive throughout its life. Notice that typically just twenty percent of the software's lifecycle cost is expended in the development phase. Further, we desire maintainability because the design engineers change over the lifecycle of a simulation. New design engineers require maintainable code to be productive. Unmaintainable software introduces a substantial learning curve resulting from the design decisions and requirements lost at the time of departure of the original designer. Finally, we desire maintainable code to improve the software documentation, which determines how well our design is understood. We can not depend on traditional documentation; in most cases it does not reflect the actual code or decisions behind the code. Maintainable code is a step toward self-documenting code.

How can we build maintainability into our design? There are four steps to implementing a design change:

1. understand all explicit and implicit design decisions,
2. design around present structure,
3. prove no error propagation, and
4. validate new feature.

Ada types directly support each step. We can make it easy to visualize ' current design by expressing the design through Ada's rich variety of types. Automatic exception checking can protect the original design from introduced errors. And, types obviously provide the same validation support to new design that they did to the original design. A good example is the enumeration type discussed earlier. If we add a new color, its location and its effect on the rest of the system is evident. The visibility into the implementation provided by Ada types is the basis for maintainability. A second powerful application of types for maintainability is variant records. With variant records, we can define a new entity in the simulation as similar to, or an extension of an existing entity. Obviously, the portions that the new and existing entities share have already been validated, which reduces the workload for the change.

Portability

Portability is the ability to transport software between computers, people, projects, and companies. Portable code is a goal of simulation that is often only considered late in the lifecycle. But early consideration of portability has a number of advantages. Portable code enables convenient platform changes. Portable code increases pro-

grammer productivity because the effort is concentrated on modeling the system instead of clever coding for the machine. Portable code reduces life cycle cost. Software costs are typically significantly higher after a simulation is fielded because of design changes (see maintainability) and equipment changes. Designs that depend on the nuances of particular machines or compilers or support tools do not hold up well over their lifespan. Also, non-portable code reduces the competitive position of an organization, which is in the position of continually redeveloping the wheel.

Ada significantly supports portability simply through its charter. Ada is a controlled, standardized language. This one fact has done a lot for portable software. Ada provides two specific type features for portability; user defined types and the extensive use of self-derived types. Through user defined types, Ada allows a programmer to define his own types' basis. This allows an engineer to remove his dependence on compiler implementation. The second characteristic of Ada dealing with portability is the huge set of new types we can introduce into the language. Enumerations, records, tasks, etc., are just a few of the examples of different types that we can use to model the simulation. With this proliferation of types, a simulation is not tied to a few machine dependent types to express its model. This volume of possible types is an advantage when producing portable code. The capability to derive and define our own set of types thus limits our risk exposure to the machine.

However, Ada has not removed all dangers. Different compiler vendors are allowed to implement fundamental subtypes under different names and sizes. Figure 7 shows an example of the names two different compiler vendors used for the various integer subtypes available. Clearly, code depending on these types would have its meaning changed as it moved between systems

Integer Width	Predefined Type Name	
	Compiler A	Compiler B
32 Bit	Long_Integer	Integer
16 Bit	Integer	Short_Integer
8 Bit	Short_Integer	Tiny_Integer

Figure 7

and might not even compile. We can avoid this loss of portability by defining our own fundamental types for integer and real numbers, and using extensive subtyping.

Reusability

Reusability is defined simply as the ability to use software again in new applications. Reusability requires portability. The potential for savings and increased profit from reusable code has provoked many studies. The benefits of reusability are not in question nor do they warrant listing. But, reusability is not as simple as it sounds. For example, restricting code to a single function does not always result in reusable code -- nor is just being generic enough. Reusable code possesses certain essential attributes such as definition of both purpose and interface. It must not be based on magic numbers. A reusable design must separate the control and the physics of a problem.

One of the most obvious ways in which Ada supports reusable code in through generics. Generics (based on types) provide the capability to develop a single algorithm for use in a wide variety of situations. One example is a user menu, which builds a menu, prompts the user, and guarantees a valid response -- one procedure for any enumeration type. Enumerations by themselves increase programmer productivity because they reduce the understandability load (trying to remember what the value 1 means here -- is it color or switch position?).

The use of attributes is perhaps the best way Ada types address reusability. When a simulation is written based on the attributes of a type, it is driven by requirements, not by parameters. The implementation algorithms can work based on the attributes of types instead of an explicit value. An example of this is aerodynamic lift parameters. Given an enumeration type defining the lift surfaces on an aircraft, then the "Compute_Lift_Characteristics" algorithm can compute lift for each surface in the type, rather than a local value. If the code is to be reused, the only change required is a modification to the list of surfaces in the type (and the database prescribing the characteristics of the surface). Any part of the implementation that considers the control surfaces is unchanged. See Figure 8 for a code example.

```
type Control_Surfaces is
 (Left_Flaperon,
  Left_Horizontal_Stablizer,
  Left_Leading_Edge_Flap,
  Left_Speedbrake,
  Right_Flaperon,
  Right_Leading_Edge_Flap,
  Right_Speedbrake,
  Rudder,
  Nosegear,
  Main_Gear);
```

Figure 8

We can further support reusability by controlling the *strength* of our types. Ada supports strong (very restrictive) and weak (non-restrictive) typing. Strong typing can insulate the data structure completely, but the misuse of strong typing can cause far more problems than it solves. We can use subtyping to express the requirement without unnecessarily interfering with data transfer in equations. Consider an equation for converting indicated airspeed into true airspeed. The equation will probably involve constants, a pressure measure, and a temperature measure. If the pressure or temperature has been created as "new" types (versus "sub"-types), Ada will not permit direct conversion. This kind of overly strong typing can cause the designer to commit all manner of bad design to work around his mistake. Properly understood, types packages are tailored extensions to the language. Anyone needing the type should have it, and the controls should be at the level of additions or revisions to the types. The rule of thumb is "all the visibility needed and no more".

Pitfalls

Ada types are not a panacea for software design. A type based implementation still requires careful design. There are a number of pitfalls that can occur with mindless typing.

1. Optimizing individual parts of a system will not result in an optimized system as a whole. We must keep the big picture in mind.

2. There is a tendency to misuse features to define complex or unusual data structures merely to facilitate a "clever coding" technique. We must avoid coding artistry but apply sound software engineering.

3. A haphazard use of types has a direct affect on the maintainability of software. We must avoid duplicating names or applications.

4. The abuse of strong typing can cause inefficient, unreadable, and unmaintainable software. Strong typing is strong medicine; we must need its power before employing it.

5. Reliance on system fundamental types is unportable. Predefined types change in both name and representation between compiler manufacturers.

6. Many times, extremely similar types will creep into the design as the software develops. This will reduce the design clarity.

7. Common global data and message passing represent extremes approaches to interface control. We should seek the balance and clarity of design which parameter lists yield.

Developers can misuse types. But we already know that we must *design* software to achieve our goals Too often in the past, engineers have given lip service to design and have then proceeded to hack out a simulation. Design is not doing things the way they have always been done. It is not making the same decisions over and over again because it worked years ago. It is employing a systems viewpoint and transforming requirements into a verified, valid, and credible model. Clearly, Ada types can play an important role in this process.

A Typing Scheme

Given the richness of the Ada typing features, we desire a consistent and unified approached to implementing types in a simulator.

1. Define your own fundamental types from the intrinsic values --
   "type Integer_32_Bit is new Integer
   range (-2**31)..((2**31)-1);".

2. Derive all of the subtypes from your new fundamental type --
   "subtype Integer_16_Bit is
   Integer_32_Bit range -32_768..32_767;".

3. Subtype wherever possible to avoid overly strong typing -- "subtype Moving_Model_Number is Integer_8_Bit;"

4. Define the interface between every object as a single type containing all of the needed information.

5. You need more enumerations than you think you do.

6. All two alternative events are not boolean (True, False) -- use enumerates as appropriate (On, Off).

7. Type names should be complete and expressive of the information -- "type Landing_Gear_State is (Locked_Up, Up, Retracting, Extending, Down, Locked_Down);".

8. Use a single package for the global simulator types at the top of the design.

9. Package the types defining the interfaces between components at a given tier in a single package one tier above the components.

10. The purpose of types is to map the design to the real world.

11. The types should express their driving requirements (as applicable), design criteria, program specification, etc.

12. Begin a project's code by prototyping the types. On the other hand, expect the types to evolve with the program.

13. There must be an owner of each types package to police additions and revisions.

14. Write code which depends on attributes instead of explicit values. Such code supports design changes simply through changing the types rather than requiring changes throughout the code.

15. Types packages, unlike other packages in the system, should be "withed" and "used" for direct access. The types are not data but a tailored extension of the language, a fundamental resource for the design.

The underlying responsibility of the types' creator is to express the design and map the simulation to the real world. Types provide a powerful simulation modeling tool, but carry a responsibility. Whereas under-typing cannot hope to fulfill the design goals, over-typing can obscure the design. A well-controlled approach to typing can produce the desired balance. However, this control depends on goodwill and agreement between team members about the typing scheme.

CONCLUSION

In the course of our experience with Ada, we have seen that designs, utilizing Ada types, correctly implemented, exhibit certain desirable qualities. Ada types can provide the glue which holds the model together. But such results are by no means a forgone conclusion. The availability of Ada types does not relieve the software engineer of the responsibility to design the product. Quality software is not the result of happenchance or black magic; it is the result of an applied software engineering approach. The developer must create a model for a set of defined goals, and then implement it with a well-defined process in order to achieve these goals. The types selected for the implementation can encapsulate the system requirements and design. Application of Ada types can be a long step toward achieving the stated goals for a simulation model. Strong inherent language features like these are a requirement for the large complex training systems being constructed today.

ABOUT THE AUTHORS

David C. Gross is a software systems engineer with the Boeing Defense and Space group in the Missiles and Space Division. He has worked in all phases of simulation and training systems from proposal through delivery. He is currently involved in basic research for software engineering and applied research for system simulation. Mr. Gross has a Bachelor of Science in Computer Science/Engineering from Auburn University and is working toward a Master of Operations Research at the University of Alabama at Huntsville.

Lynn D. Stuckey, Jr. is a software systems engineer with Boeing Defense and Space Group in the Missiles and Space Division. He has been responsible for software design, code, test, and integration on several Boeing simulation projects including the Ada Simulator Validation Program and the Modular Simulator System. He is currently involved in research and development activities dealing with software development for weapon and threat simulators. Mr. Stuckey holds a Bachelor of Science degree in Electrical Engineering from the University of Alabama, in Huntsville and is working toward a Master of Systems Engineering at the University of Alabama, in Huntsville.

THE CHALLENGES OF DEVELOPING A REAL-TIME ENVIRONMENT IN ADA

Walter E. Zink, Sr., and Jill M. Neebe
CAE-Link Corporation
Binghamton, NY

ABSTRACT

With fewer and fewer exceptions, the Department of Defense is requiring Ada to be the sole programming language for all new software-related projects. In addition, these new projects are expected to achieve higher levels of maintainability from a software perspective. Experience shows that these seemingly unobtrusive requirements manifest themselves in a very large variety of unforeseen challenges and implicit requirements. This paper overviews an Ada real-time flight simulation environment based on an implementation for the B-2 Aircrew Training Device (ATD) and the challenges encountered in going from concept to product. Three areas of challenge are involved in building this environment. The first concerns the control of software units distributed across processors and groups of processors. Another area of concern is providing input/output services to all units in Ada, which even the operating system does not readily support. The third area covers selected obstacles encountered in developing a pure Ada implementation of a system to support unit interfaces. The resultant real-time environment represents an effective blend of Ada and traditional techniques.

INTRODUCTION

The Simulation Environment discussed in this paper is a Real-Time Simulation Environment (RTSE) in which the application/subsystem software executes. It is divided into three principal areas: control, software interfaces, and input/output (I/O) (Figure 1). In this context, control consists of implementing the current user's instructions and orchestrating the execution of other simulation software.

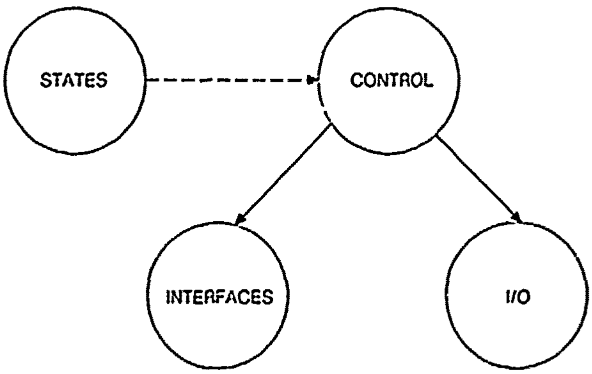


Figure 1 Real-Time Simulation Environment

Since Ada is the language of the RTSE and DOD-STD-2167A is a currently used standard for Ada programs, DOD-STD-2167A phases will be used to chronologically orient the requirements and implementations. Figure 2 outlines these for each product area. The discussion covers System Requirements Analysis to Coding and Computer Software Unit Testing.

While Ada has attractive capabilities not available in other languages, compiler capabilities can dictate IF and HOW various features can be practically implemented. The Ada Joint Program Office, a part of the United States government, sets the criteria and certifies Ada compilers. In the 1985/1986 time frame, Ada compilers were still in their infancy. The Ada Programming Language Military Standard, ANSI/MIL-STD-1815A, was approved in February 1983. Only three compilers had received validation by December 1983, and in 1985 only 14 validated compilers were available on the market.[1] The validation criteria did not require all of the features of Ada needed by a RTSE.

The majority of previous simulation software produced by the company consisted primarily of FORTRAN code; the B-2 ATD was the first "Ada simulator" for the company. Previous design experience was primarily functional decomposition. With Ada we use Object Oriented Design techniques. New procedures needed to be developed and written from an Ada perspective.

Naturally, Ada concepts played an important part in RTSE implementation decisions. As stated in MIL-STD-1815A, "the development of programs is becoming ever more decentralized and distributed. Consequently, the ability to assemble a program from independently produced software components has been a central idea in this [Ada's] design. The concepts of packages, ... and of generic units are directly related to the idea."[2]

2167A PHASES →	SYSTEM REQUIREMENT ANALYSIS/DESIGN	SOFTWARE REQUIREMENTS ANALYSIS	PRELIMINARY DESIGN	DETAILED DESIGN	CODING & COMPUTER SOFTWARE UNIT TESTING
CONTROL	PORTABILITY			MACHINE DEPENDENCIES ISOLATED	
	REUSABILITY		USE GENERICS AND PSEUDO_GENERICS		
	MAINTAINABILITY		USE GENERICS AND PSEUDO_GENERICS		
		PROVIDE A CYCLIC ENVIRONMENT TO RUN USER SOFTWARE	EXECUTIVE DESIGN		
		PROVIDE CENTRAL CONTROL ACROSS PROCESSING ELEMENTS	MASTER/CLUSTER/ SEQUENCER CONCEPT		
		PROVIDE AUTOMATIC LOADING LEVELING	USE OF MULTIPLE-RATE O/S PROCESSES		
		PROVIDE SINGLE-POINT CONTROL		BOOTSTRAP	
		MINIMIZE OVERHEAD ASSOCIATED WITH THE REAL-TIME ENVIRONMENT		O/S PROCESS CONTROL IN CLOCK HANDLER	
INTERFACES	REUSABILITY				
	MAINTAINABILITY	MINIMIZE SYSTEM-WIDE RECOMPILATION OF APPLICATION SOFTWARE		WITH AT BODY LEVEL OF CONNECTION MANAGERS	
		PROVIDE AN AUTOMATIC METHOD TO MANAGE THE INTERFACES, GENERATE THE INTERFACING SOFTWARE AND DOCUMENT THE INTERFACES	SHARED MEMORY MANAGEMENT - INTERFACE MANAGEMENT DATA BASE	DATA BASE ACCESS SOFTWARE INTERFACE GENERATION	
	INFORMATION HIDING				
	DATA ABSTRACTION				
			ENSURE DATA CONSISTENCY FOR INTERFACES	IMPORT/EXPORT CONNECTION MANAGER	
			ENSURE DATA INTEGRITY FOR ALL INTERFACES	DOUBLE BUFFERING OF GLOBAL MEMORY	
		MINIMIZE THE OVERHEAD ASSOCIATED WITH INTER-SOFTWARE COMMUNICATION			
		PROVIDE THE CAPABILITY TO SAVE THE SIMULATOR ENVIRONMENT AND RESTORE THE ENVIRONMENT FOR EITHER A POWER FAILURE OR AN INSTRUCTOR'S REQUEST		SNAP/RESET PRODUCT	
		PROVIDE INTERFACE BETWEEN SOFTWARE COMPONENTS WITHIN THE SAME O/S PROCESS AND BETWEEN O/S PROCESSES EXISTING ON CLUSTER OF PROCESSORS		GLOBAL MEMORY IMPORT/EXPORT CONNECTION MANAGER	

Figure 2 Requirements and Implementations

2167A PHASES →	SYSTEM REQUIREMENT ANALYSIS/DESIGN	SOFTWARE REQUIREMENTS ANALYSIS	PRELIMINARY DESIGN	DETAILED DESIGN	CODING & COMPUTER SOFTWARE UNIT TESTING
I/O	PORTABILITY		USE OF ADA CONSTRUCTS OF TEXT_IO AND DIRECT_IO	MESSAGE_REQUEST OPERATIONS, DIRECT_IO OPERATIONS	
	REUSABILITY		USE GENERICS	REQUEST_DIRECT_IO, REQUEST_CONTIGU-OUS_IO	
	MAINTAINABILITY		USE GENERICS: USE ENGLISH-LIKE NAMING PRACTICE	REQUEST_DIRECT_IO, REQUEST_CONTIGU-OUS_IO	
	DATA ABSTRACTION		USE PACKAGES	MOST I/O SOFTWARE	
	OUTPUT MESSAGES TO ERROR LOGS, AUDIT LOGS, AND TERMINAL IN SPECIFIC FORMAT			MESSAGE_REQUEST	
		NO-WAIT I/O		INTERFACE STRUCTURE. ALL USE MODULE REQUEST SOFTWARE. ALL I/O PROCESSOR SOFTWARE	
		I/O ACROSS PROCESSORS		INTERFACE STRUCTURE	
		I/O ACROSS CLUSTERS		INTERFACE STRUCTURE	
		I/O FOR OBJECTS GREATER THAN 64K		REQUEST_CONTIGU-OUS_IO, CONTIGU-OUS_IO OPERATIONS	
		DISK MARKING		REQUEST_DISK_MARKING	

Figure 2 Requirements and Implementations (Cont'd)

We found the packaging concept advantageous from a design perspective. The concepts of limited visibility and information hiding supported our requirement for data abstraction, as did the ability to logically group types and objects with applicable functions and procedures. Packaging provided us with a means to partition the program software. The ability to overload functions and rename adds valuable flexibility.

The more reusable a module of code, the lower the actual cost per line of executable code. Generics are one method of avoiding repetitious code. This reduces software testing and configuration management complexity. Since Ada does not support passing of packages as formal generic parameters,we used package renaming and software built from a common template.

Ada provided tasking; our tasking experiences are discussed in relation to the control and I/O sections.

SYSTEM LEVEL DESCRIPTION

States Overview

A state design is the basis of the structure of the Real Time Simulation Environment (RTSE) software. A state is defined as follows:

A state is a collection of rules within which a functionality can occur. The transition to another state is predicated upon successfully satisfying the state transition rules.

One implication of this definition is that the software executing in a particular state shou d be tailored to support only that state. This concept is in contrast with earlier implementations wherein the user software supported all states and had to interrogate multiple flags in order to determine appropriate functionality. In the state design used, the high level software functionality is determined off-line rather than during execution.

Another implication is that the definition of a command in one state does not have to be the same in a different state.This implication is observable in the functionality of the Control section. When the Configuration_State is Integrated, the definition of the command "Normal Load" is to load all clusters; however, in the Stand_Alone Configuration_State, the definition is to load only the cluster that has received the command.

The RTSE can be thought of in terms of three levels of states. States are supported directly by the structure of the Control section. The three state areas and their compositions are depicted below.

States	Super States	Configuration States	Element States
State Compo-sition	Pre-Synchronous (Boot Up)	Integrated	Initialization Freeze Real Time
	Synchronous	Stand_Alone	Reset
	Post-Synchronous (Shut Down)		Test

The aggregate of pre-synchronous activities is referred to as Boot Up. Post-synchronous activities are referred to as Shut Down. During Boot Up, the entire simulator is powered up, software loaded and started. During Shut Down, the entire system is stopped, cleared, and powered down. The Synchronous Super State is where the majority of the user software is cycled. It is in this Super State that the Configuration_States and Element_States have the most influence on the operation of user software.

The two Configuration_States are Integrated and Stand_Alone. Integrated means all clusters are synchronized and communicating among themselves. This refers not only to user level software, but also to RTSE level software (e.g., Control and I/O). In the Stand_Alone state a cluster functions in isolation from the other clusters.

Element_States primarily affect the execution of the user level software only and do not possess the systemwide influence of the Configuration_States. The Element_States are implemented by the Control Section via Control Records. Since each of the two Configuration_States supports the five Element_ States, there are ten Control Records.

Hardware Overview

The hardware configuration for the simulator can be partitioned into the following basic building components: the Basic Simulator Unit (BSU), the Instructor's Station, a visual system, and five multiple target system computer clusters. Since most of the real-time software executes on these computer clusters, a brief description of that hardware follows (Figure 3).

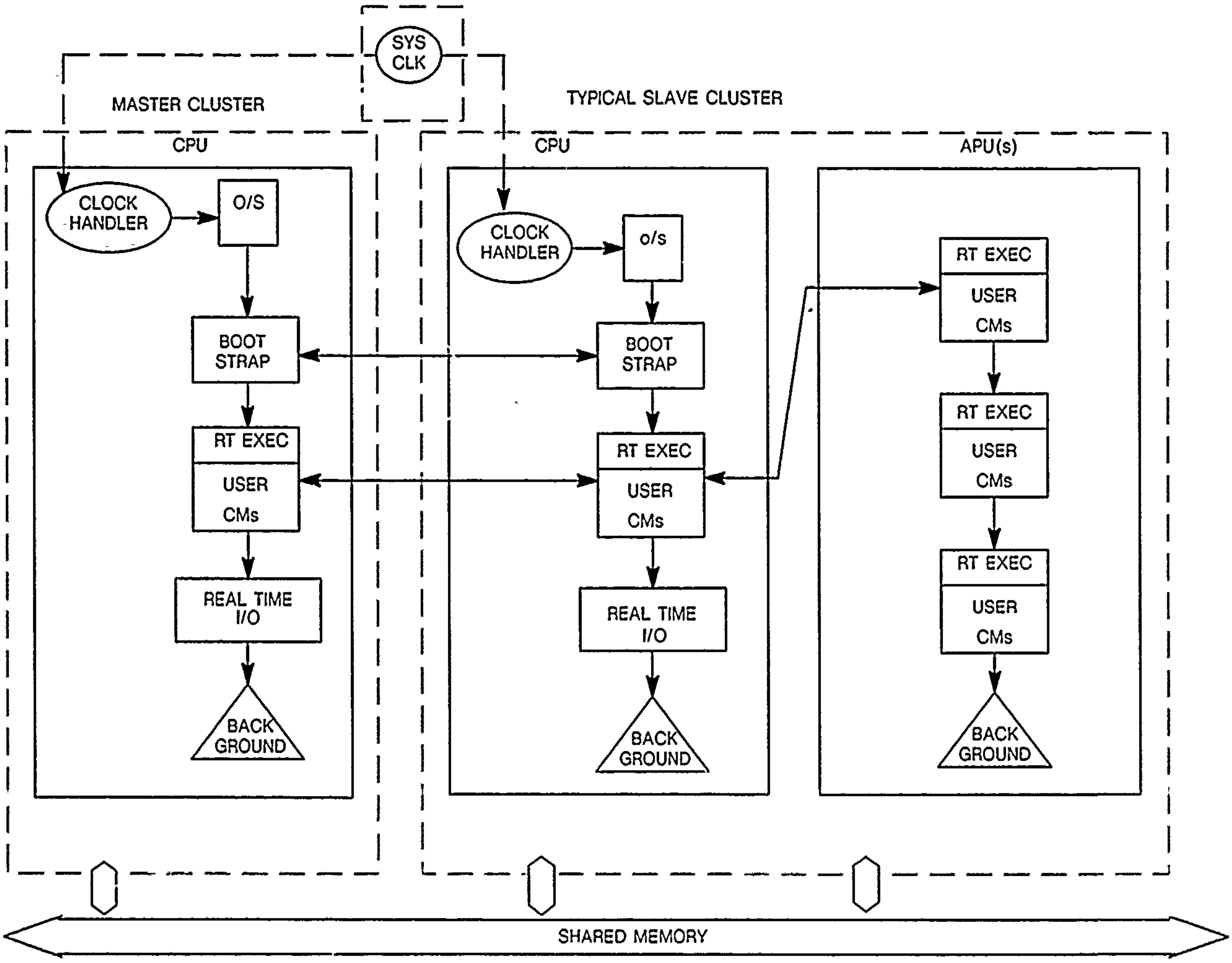


Figure 3 Hardware/Software Architecture Overview

Each cluster contains a Central Processing Unit (CPU), zero to three Auxiliary Processing Units (APU), and 32 megabytes of memory. Intra-cluster (i.e., between a CPU and one or more APUs) communication is performed through a part of memory which has local (to the cluster) visibility only. Inter-cluster (i.e., among multiple clusters) communication is through a special part of the memory that is visible to more than one cluster. In addition to data communication, all clusters are synchronized at the major cycle boundary. The System Clock (Figure 3) issues a hardware interrupt to each cluster once every major cycle, (refer to Figure 4). At this point all clusters will begin execution of the first frame of that cycle together.

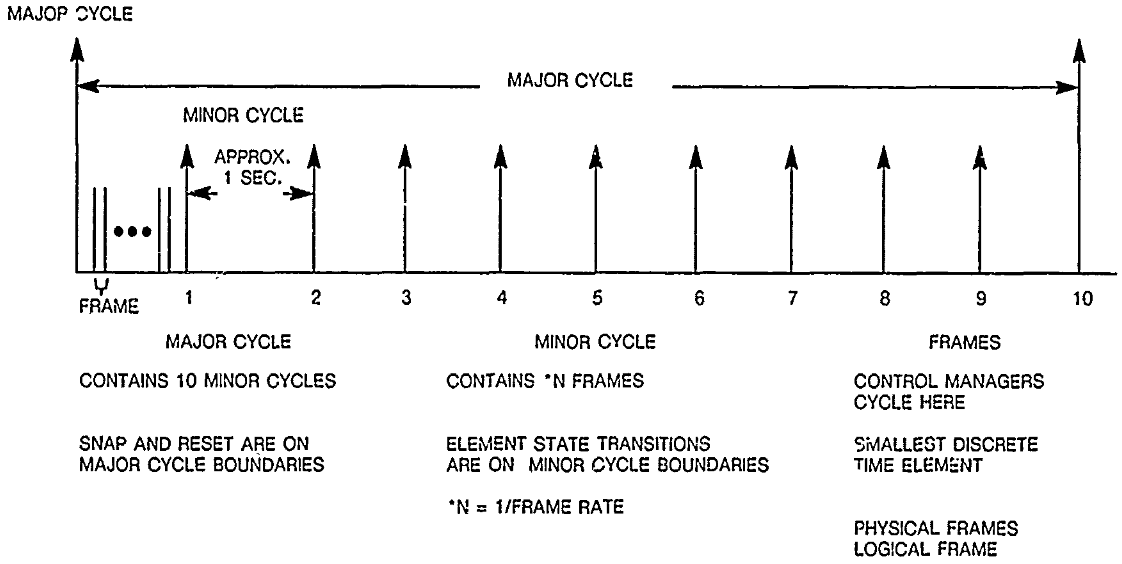


Figure 4 Frames/Minor Cycles/Major Cycles

Software Overview

The software for real-time execution is divided into three levels: Operating System level, RTSE level, and User (i.e., application) level (Figure 5).

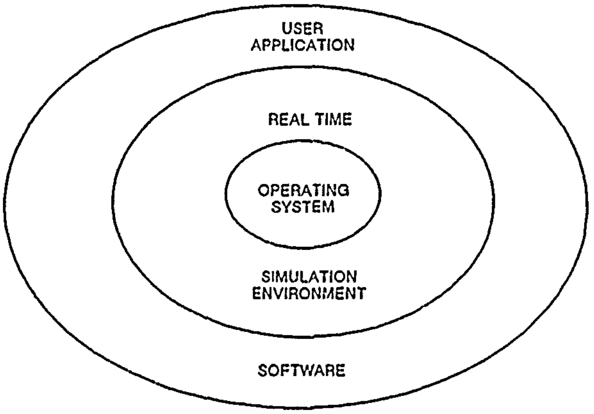


Figure 5 Software Hierarchy

The operating system level software consists of all vendor-supplied O/S services and any application-unique services such as the clock handler, which provides the connection between the hardware framing interrupts and the Control section. The principal activity of the O/S is to handle the scheduling of os_processes. This activity includes scheduling of os_processes on the same processor with different priorities as well as scheduling os_processes on different processors. The os_process with the highest priority is scheduled to run first by the O/S. When that process gives up control, the O/S will schedule the os_process with the next highest priority, and so forth. The RTSE uses this method to achieve auto-load balancing and frame-bound execution at the same time. The other services that the O/S provides are low-level interface software to hardware devices such as terminals and disk drivers.

The RTSE software is the next level of software above the operating system level. This software encapsulates the users' application software, isolates it from hardware constraints, and provides the structure and form necessary to implement the architecture. This level of software consists of the following software products which fall into three groups:

Groups	Control	Interfaces	Input/Output
Products	Executive Bootstrap	Shared Memory Management	Real-Time I/O

In addition, there is support software that is used during off-line development for the Executive, Real-Time I/O, and Shared Memory Management.

The Executive is the central component of the RTSE level of software. Although most subsystems (user software components) within the simulator are assumed to be autonomous, most of the RTSE level software is related to the Executive. The Executive schedules user software as a function of the cluster,

processor, state, and frame. User software is also referred to as Control Managers (CMs). Simulator states define the current functionality. Framing provides discrete intervals in which part of an activity must complete (e.g., integrating distance as a function of airspeed). The state in which the executive is running determines the personality of the entire simulator, even though other software (e.g., Bootstrap, Instructional Station user software) can alter the state of the executive.

The executive comprises three levels of control: master, cluster, and sequencer. Only one Master Executive exists. It is the highest level of control and manages all simulator common activities, such as fielding state change requests while in the integrated state. The Cluster Executives (one per cluster) directly or indirectly schedule all synchronous software within a cluster, provide physical and logical framing for the user software, and provide local management of states. The sequencer executive is the lowest level of executives and does the actual scheduling of the user software.

Bootstrap loads all simulator-related software and provides some control functions/utilities such as memory clear. The functionality of Bootstrap is a direct consequence of the configuration state of the executive.

Real-Time I/O provides the user software with an Ada-like interface for requesting disk and terminal services, and provides for error/audit logging. The functionality of Real-Time I/O is likewise dependent on the configuration state of the executive.

Shared Memory Management consists of two broad functions: a real-time operation and a database management function. The real-time function is to provide inter-user software communication via import/export connection managers and to provide Snap and Reset functionality.

The Real-Time library is a set of standard mathematical functions for integer, float, and long float operations that the user software can reference.

The user software is the last level of software in the simulator. This level consists of simulation-specific software.This software is broken down into subsystems and each of those is further divided into one or more control managers. Each subsystem is encapsulated with an import connection manager (IMP) and an export connection manager (EXP) (Figure 6). These connection managers are a part of the Shared Memory Management software.

CHALLENGES ENCOUNTERED

Control

Requirements and Challenges - The requirements used to design software can be organized into two principal categories: those of a product-specific nature and those of a software engineering nature. Often the requirements from one category will conflict with the optimum method of implementation of a requirement from the other category.

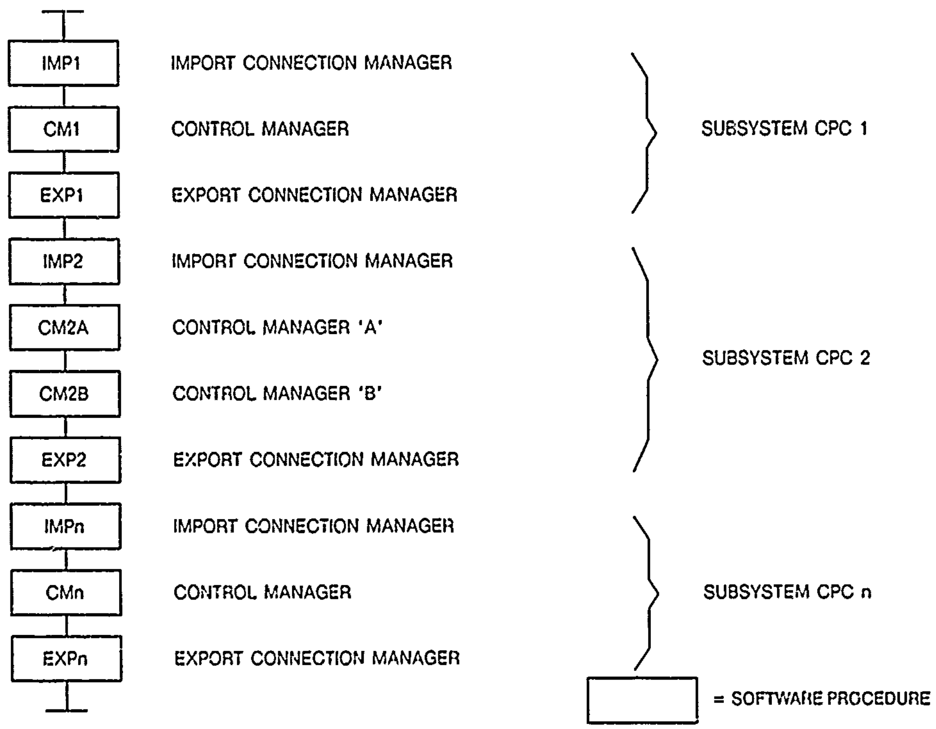


Figure 6 Typical Subsystem Execution Sequence

The product-specific requirements used to define the control portion of the real-time environment are fairly similar to those of previous simulators. They are as follows:

a. Provide a cyclic environment in which to run user software.

b. Provide central control across processing elements.

c. Minimize the overhead associated with the real-time simulation environment.

d. Provide automatic load leveling.

e. Provide single point control.

f. Provide a state-driven environment to envelope the user software.

The "Software Engineering" requirements are as follows:

a. Make the product portable.

b. Use as many reusable components as possible.

c. Make the product easy to maintain.

d. Maximize data and control abstraction.

The following is a discussion of each product-specific requirement and the software engineering requirements which affect it. The requirement for a cyclic environment is essentially a practical decision. There are basically two ways to handle asynchronous events. The first method is to run a cyclic process fast enough to service the event without a noticeable effect on event response time (such as changing a switch position, observing when a light comes on, or some other hardware event). The other way is to establish a system of interrupts for the asynchronous events and interrupt handlers to interface with the event processing software. All of this is then wrapped in an Ada tasking construct. There are three reasons for not choosing the Ada tasking/interrupt approach. First, high context switch times were witnessed during our benchmark testing. Second, risk was involved in mapping multiple hardware interrupts to user level software. And third, some iterative processes still had to be calculated cyclically.

The requirement for central control across all processing elements is a departure from previous endeavors where control is more regional in scope. Unlike the previous requirement, this one lends itself well to various software engineering attributes such as packaging, data abstraction, and information hiding. The requirement is implemented via the three-tier approach to the executive. When the simulator is integrated, the master executive manipulates data that is common to all clusters (such as shutdown state changes). This data is in a separate package which only the master executive writes. Each cluster executive then reads the system level data and controls its cluster accordingly. Similarly, the cluster executive manages data which is common to all sequencer executives in the cluster. This concept is taken one step further by sequence executives which manage all environment-related data used by the application software running under them. This concept of partitioning data into separate packages takes advantage of two of Ada's built-in software engineering concepts: data abstraction and information hiding. The requirement to provide central control is not limited to the management of common environment data. The cluster executives control the execution of all sequencer executives in their cluster. The original implementation was accomplished by:

a. Setting the time_to_executive flags

b. Monitoring Frame_complete flags

c. Controlling subordinate sequencer executives, executive os_processes, via an Ada interface to the operating system.

These activities use standard Ada programming constructs. The Ada interface to the O/S dependent software is localized to a single package to enhance transportability. However, controlling an os_process via an Ada interface to the O/S proves to be an exorbitantly time-intensive activity. Depending on the number of APUs to be rescheduled and the number of lower rate CPU executives, 25 to 35 percent of the cluster executive's frame time would be consumed by this activity. This directly conflicts with the requirement to minimize overhead. The resulting decision placed the functionality of subordinate sequencer executive's control into each cluster's clock handler. This decision reduces the activity's execution time by up to sixfold! Even though the clock handler is an assembly language program, all of its data (i.e., os_process names, etc.) is still managed by the RTSE, and thus retains much of the original Ada software engineering attributes.

The requirement to provide automatic load leveling is met by the use of multiple executive os_processes running on the CPU and APUs. Although the actual code implementing the First In First Out (FIFO) queues on the APU is O/S dependent, the system design and structure is portable since most computer operating systems provide a mechanism to accomplish this activity. Since the O/S-dependent code is isolated in one package, it can be conveniently replaced by the appropriate interface software when transporting the design to another system. The disappointing reality is that at the time

these load leveling decisions were being made, current implementations of Ada tasking did not support this requirement. In theory, Ada tasking should support a load leveling activity for a single processor. Our initial benchmark test results of Ada tasking in the 1986 time frame did not support the Ada tasking implementation we had expected. We observed that the entire os_process paused if even a single Ada task within that os_process paused. We had expected the pause to cause a context switch to another Ada task. In addition, the Ada task context switch took approximately as long as an os_process context switch. It was anticipated that Ada tasking would be much quicker since it was strictly part of the application software and did not depend on an operating system interface. Ada's failure to recognize multiprocessor cluster is a fundamental deficiency in an otherwise remarkable language.

The requirement for single-point control is considered to be relatively uncontroversial. In earlier simulators, with fewer computer hardware components, the process of bringing up each cluster individually was relatively simple. However, this simulator was considerably more complex and the process would have been very detailed and involved. Thus, this requirement was reasonable and logical. For hardware reasons, the design (i.e., the bootstrap product) to accommodate this requirement uses multiple, nearly identical os_processes across the simulator. These os_processes differ only by the actual data package which contains the cluster's unique personality. In light of the requirement to make our products more reusable, the use of a generic main procedure would have been the preferred implementation. However, Ada does not permit the passing of a package as a generic formal parameter. Therefore, the pseudo-generic technique of package renaming was used. The product software was identical in all cases, thus adding to the maintainability. The local version of each cluster-specific package is referenced throughout the bootstrap product. Each version of this software contains a rename of a cluster-specific package to a local package name. Since the cluster-specific package is not in the direct withing structure of the main, the main is the same in all clusters and meets the criteria of reusable software. However, a separate library or sublibrary must be maintained for each cluster of this software because of the renaming of the cluster-specific data package to the local package name. This approach poses no problems in the design and development phase; however, the use of multiple libraries/sublibraries complicates load production due to library management, compilation, and linking issues.

This real-time environment had to support a simulator that was roughly an order of magnitude larger than previous ones, and the old software control methods were not sufficient. The requirement to be a "state-driven" environment eliminated some of the size-related issues and provided a mechanism to actively support control abstraction. This state-driven requirement addresses the overhead issue as well as size and control abstraction. As previously stated, the software was tailored to support only those activities related to the state in which it runs. At first, it was feared that this approach would greatly increase the amount of the user application software because there would be many copies of nearly redundant code. However, Ada packaging can be used to maximize utility of a state-driven executive, if the software is organized by the functionality it must support in any given state. There are three steps involved in this process. The first is to organize (via packaging) the software of a particular operation (e.g., an engine model) into the smallest identifiable operations, known as primitives. The second step is to organize the primitives into groups that support the required functionality in a given state. For example, all primitives that are involved in integrating quantities would be included in the group of primitives that execute in real time and would not be included in the freeze group of primitives. These groups are simply Ada procedures referred to as Control Managers (CMs). The third step is to identify to the executive the applicable execution state for the CMs.

This approach has a twofold advantage. First, the decision as to when software executes is made off-line during development and does not contribute to the real-time overhead. The second advantage is that only software applicable to the current state executes. For example, a popular method of summing a value in real time is to calculate a delta value, scale that value, and add it to the running total. In previously used approaches, the scaling constant would have two values: one for real time and another for freeze. The real-time scaling constant would equal the inverse of the frame rate and the freezing constants would equal zero. The software would always execute and simply change scaling constants to implement real-time or freeze. This is not the case in the state-driven approach. In our state-driven approach, the summing procedure would only run in the real-time state and would not be called in the freeze state. The states in which a procedure is to run is listed in the Configuration Control File (CCF).

The system definition for the entire simulator is contained in the CCF. This file correlates the user

software (Control Managers - CM), states in which each CM will execute, and which executive os_process contains the CM.

Implementation - As previously mentioned, the executive steps through the three superstates: pre-synchronous, synchronous, and post-synchronous. During the pre-synchronous superstate, each executive performs the user initialization and the executive initialization. The user initialization procedure provides a standard interface to the Executive for the user software. Within the initialization superstate, all necessary user-defined procedures are called to perform initialization for the users. Once a particular Executive completes its initialization activities, it waits for all other Executives to complete their initialization before entering the synchronous superstate. The waiting is accomplished via rescheduling for APU executives, suspends for CPU sequencer executives, and a clock interrupt for master/cluster executives.

After entering the synchronous super state, the following functions are performed by the Executive:

- Timing;
- Frame and Cycle Boundary Updates;
- Scheduling of OS Processes and User Procedures.

If the timing of frames has been requested, the real-time clock is read at the beginning and end of the frame and statistics are kept.

The master/cluster Executive keeps track of physical frames, cycles, and changing of states. Once the frame and cycle boundary logic is complete, the master/cluster executive manages all sequencer executives within the cluster. Each sequencer executive calculates its own logical frame number. Then the sequencer executive calls the user procedure (CMs) based on the state and logical frame number. If requested, the sequencer executive times user procedures and maintains the statistics.

Interfaces

Requirements - On previous simulators, information was shared through a construct known as a global data pool. These constructs were subject to data inconsistency, unnecessary global visibility, and data corruption.These factors drove the development of the requirements for the interface software.

For interface software, we observed very little conflict between the product-specific requirements and software engineering requirements, because this product is system level in nature. The product-specific requirements are as follows:

a. Provide interface between software components within the same os_process and between os_processes existing on a cluster of processors.

b. Provide the capability to save the simulator environment and restore the environment for either a power failure or an instructor's request.

c. Minimize the overhead associated with inter-software communication.

d. Provide an automatic method to manage the interfaces, to generate the interfacing software, and to document the interface.

The software engineering requirements are as follows:

a. Ensure data integrity for all interfaces.

b. Ensure data consistency for all interfaces.

c. Minimize systemwide recompilation of application software.

d. Actively support data abstraction of the interface objects.

e. Actively support information hiding of the interface.

f. Design the product to be maintainable and reusable.

The requirement to provide interfaces between software components within the same os_process and between os_processes existing on a cluster of processors has a far-reaching effect on the RTSE design. Due to the large size of this simulator, it was immediately apparent that this simulation would span many os_processes and processors. A method to handle the interfaces had to be established. The first task was to identify hardware which supported the software resource needs. The options were to find either a single memory in the 256 to 512 megabyte range or an arrangement of interconnected memories which function as a single memory. In conjunction with the hardware requirement, we needed a compatible validated Ada compiler to be paired with it. In our early 1980's study, the coupling of these two requirements effectively reduced the field of viable vendors to one.

The vendor-supplied interface mechanism is at the Ada package level and uses the construct of a "Task Common" area (TCOM). Multiple Ada packages can be linked to a particular TCOM without any coding changes to the Ada packages. This preserves the original data abstraction designed into those packages. All os_processes linked with a par-

ticular TCOM could have visibility to any data if they withed the applicable package on the TCOM. This method preserves Ada's visibility rules. Through this mechanism, software units can interface with other software units anywhere in the system using normal Ada constructs. This method is used by all three areas of the RTSE.

The interface software requirements tend to reinforce one another, with the requirement for low overhead being the only exception. This is a distinct contrast from the control software requirements. Although the interface requirements are basically symbiotic, many challenges arose during the implementation, most stemming from the sheer size of the simulator software.

The RTSE maintainability requirement encapsulates the entire product and was coupled with the requirement to automate the generation, management, and documentation of interface connections. The Interface Management Data Base (IMDB) and the supporting software were the result. The IMDB and supporting software generates interface packages through an automated, menu-driven process. Our system currently manages approximately 10,000 interfaces. It also automates the process of ensuring data consistency between the interfacing components. Since this automated process is the only method for generating or changing interfaces, no informal modifications can be made to the interface software, hence protecting the integrity of the interfaces. Once the IMDB and associated software have been through testing, generated packages need not be tested. The reusability of the IMDB and associated software in this fashion supports maintainability of future interfaces or changed interfaces providing effectively pre-tested packages to the user.

Although understood by most application software engineers, the concept of the FORTRAN data pool and its symbol dictionary was simply unfeasible for a number of reasons. The first reason is that the number of interfaces is many times that of previous simulators. Any attempt to manually manage that unwieldy number of interfaces was impractical. The Ada packaging capability, coupled with the requirement for information hiding, assisted in the solving of the interface problem. Packaging under Ada allows related objects to be grouped together in packages small enough to be meaningful and managed at all levels of integration. Data integrity is greatly enhanced by use of Ada packaging. Unintentional corruption of data is practically impossible with Ada because one must consciously "with" a package in order to alter any of its contents. These packages are generated from the IMDB and the Configuration Control File (CCF).

Another challenge was the view, entertained by some, that the benefits of packaging and information hiding did not fully compensate for the loss of flexibility to change data especially in a test environment. In certain dynamic software tests (e.g., Air Vehicle modeling), large numbers of objects are manipulated. However, this situation was anticipated and a separate product (CoASTE - Coherent Automated Simulation Test Environment) outside the realm of RTSE was provided to support this activity using primarily standard Ada constructs.

The views on the level of abstraction ranged from permitting only three types — Integer, Float, and Boolean — to "if Ada supports it, then use it." The final decision was to support data abstraction to the maximum extent possible with only a few restrictions. The principal restriction limits the full path name of an object to 112 characters in length.

The final challenge was the overhead issue. An approach which ensures data integrity and consistency may use more overhead than one that does not provide those services. The overhead consists of execution time and memory allocation. For a software project this large, it was felt that data integrity and consistency could not be sacrificed in favor of a low overhead requirement. To minimize the impact of increased overhead, several things were done. To lessen the impact on any individual, the overhead of ensuring data consistency and integrity was distributed over both data importers and exporters. Activities such as constraint and range checking could be employed during product development and validation, and then turned off (i.e., compile without them) to meet the spare time requirement. The use of a double buffering scheme was implemented to protect data integrity and ensure data consistency.

<u>The Design</u> - To accommodate the many design requirements, one off-line and two real-time products were designed. The off-line product was designed to manage and document the software component interfaces as part of an integrated load build system. The actual automatically generated (auto-genned) interfacing software and the software to do the capture and restore comprises the real-time software. The collection of these products is known as Shared Memory Management (SMM).

The off-line software incorporates the use of an Oracle database to manage and document the interfaces and to build the interface software. This part of SMM consists of three sections. the Interface Management Data Base (IMDB), Data Base Access (DBA), and Software Interface Generation (SIG).

The DBA section consists of all software that is needed to insert or extract database information. DBA is written entirely in Ada and uses the Oracle provided interface to the database. Access to the IMDB for entering and deleting information consists of two levels of activities, Administrative Tool Package (ATP) and the Element Tool Package (ETP). Paper forms are used to describe the software interfaces changes and are processed via ATP and ETP on the IMDB.

The ATP permits the IMDB Administrator to insert, update, and delete IMDB entities via the IMDB ATP menus. It is the IMDB Administrator who controls all direct IMDB change activities (via the IMDB information forms and the element auto-entry text files), while both IMDB Administrator and users (ETP) are granted privileges to generate IMDB reports.

Also, the ATP will provide the IMDB Administrator with the ability to extract information from the IMDB and generate connection manager packages (Import and Export/State), shared data packages (Shared Memory), and various reports.

The user requests IMDB changes through the ETP. While the menu choices are similar to those of the ATP, the user's changes do not go directly into the IMDB as ATP changes do. User changes are stored in an auto-entry text file. The IMDB Administrator then reviews the changes, and if approved, processes them into the IMDB. The ETP allows each user to generate various reports from the IMDB. It is this part of the SMM IMDB process (i.e., IMDB information forms and element auto-entry text files) that is solely under control of the users.

The users must define all export objects relative to each control manager in the IMDB. A logical group consists of an importer, one or more control managers, and an exporter. Logical groups (Figure 6) are defined in the Configuration Control File (CCF).

The last part of SMM DBA is the extraction function. Each import CM, export CM, and applicable state data is associated with a logical group. An information list for each is extracted from the IMDB based on the definition of the logical group as defined in the CCF. These information lists serve as input to the Software Interface Generation (SIG) process.

The Software Interface Generation (SIG) process will generate shared data packages (i.e., Shared Memory) and connection manager packages (i.e., Export/State CM packages and Import CM packages). During software generation of connection managers packages and shared data packages, all non-interfaced export and/or import objects will be filtered out. The information for this process is contained in the information lists provided by the DBA extraction function. The resulting software interface product consists of CMs and related shared memory packages.

The connection managers provide inter-simulation communication and execute in real time as part of the executive in the same fashion as the control managers.

The connection managers include code to import and export inter-simulation communication variables (shared objects). They also include code to transfer state data between local data area and a shared data area for the snap/reset process. Also included in the connection manager is software to initialize the associated "shared memory" to values derived from the IMDB. Since connection managers are built software, each connection manager has the same format.

Each export connection manager has a single shared data package associated with it. The shared data package contains a type package defining a record containing all of the variables to be exported and a current side of memory pointer. This record is a "side of memory". An array of two of these records is then used to provide two "sides of memory" so that one "side of memory" can be active (i.e., exported to) while the other "side of memory" remains constant. Therefore, the snap process reads valid, consistent (with respect to time) data from the other "side of memory", thus providing doubled buffering of the data.

A second package, the "shared memory" package, contains an object of the array type defined in the type package. Each side of shared memory is separated into export data and state data. State data is information needed for reset/restore. Shared memory uses buffering techniques, where necessary, to manage data consistency among sets of export data being stored. In such cases, multiple buffers of export data are maintained, avoiding simultaneous reads and writes to a single area (Figure 7). The third and final package contains the exporter, the initialization for the "shared memory", the state data export, and the state data import.

The import side of the connection manager contains only one package. This package has an import procedure that transfers "shared objects" from one or more "shared memories" to the user's software local data area.

The import procedure will use user-defined filtering techniques (e.g., interpolation and extrapolation), where necessary, to manage data consisten-

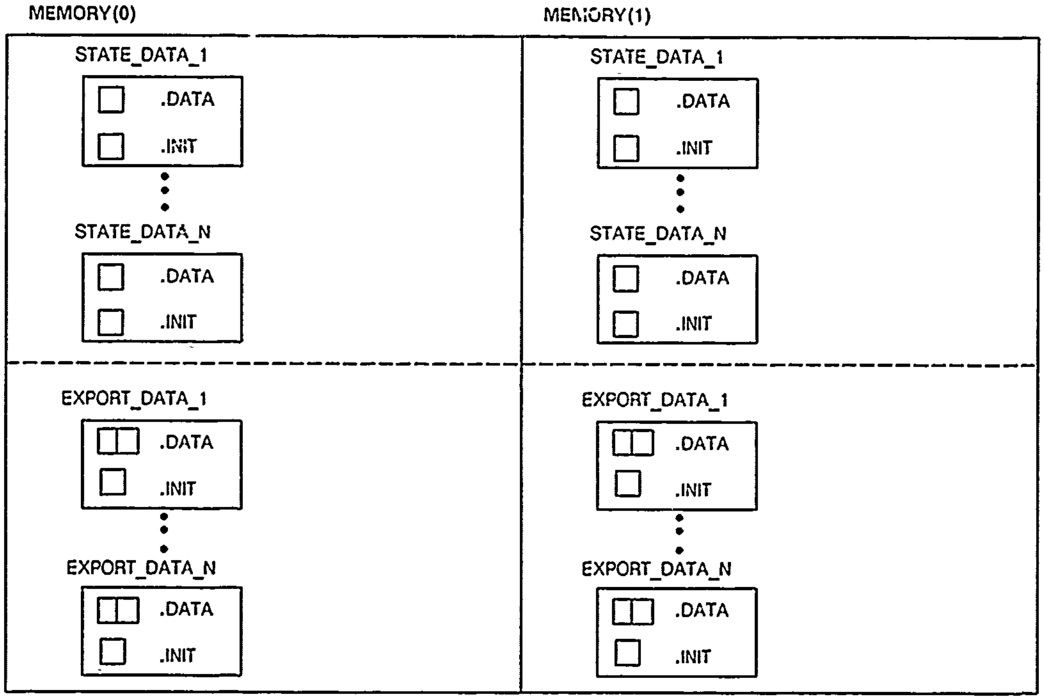


Figure 7 Shared Memory Diagram

cy among sets of import data moved from the shared memory areas.

The following is a description of the snap/reset software used to implement the requirement to save the simulator environment. During a power fail snap or an instructor requested snap, the current side of shared memory pointer is toggled. During a snap, one side of memory will be written (i.e., saved) to disk for later use in the reset/restore process, while the other side of memory becomes active to support importing/exporting. The side of memory being saved to disk will not be updated until the save is complete. A reset request to a previous point in time is the reverse of the above process.

Real-Time I/O

Requirements – The I/O requirements are divided into two groups: Software Engineering requirements and Product Specific requirements. Each group of requirements will be listed, followed by detailed information on each Product Specific requirement. Since the Software Engineering requirements generally support or conflict with the Product Specific requirements, these relationships will be covered in the Product Specific requirements discussion. Software Engineering requirements which are not fully addressed in the other sections are grouped at the end.

The Product-Specific requirements are as follows:

a. No-Wait I/O

b. I/O Across Processors

c. I/O Across Clusters

d. Disk I/O : direct_io

e. Disk I/O for objects greater than 64K

f. Output messages to error logs, audit logs, and terminal in specific format

g. Disk Marking

The Software Engineering requirements include the following:

a. Portability

b. Reusability (Internal and External)

c. Maintainability

d. Data Abstraction

Challenges/Conflict Resolution - The following challenges/conflicts had to be met/resolved:

a. No-Wait I/O

A real-time environment requires no-wait I/O. Our real-time environment's integrity could not withstand the delays of Ada-provided I/O constructs. No-wait file I/O in our context does not mean that the actual I/O happens at the instant requested but rather that the requesting software need not wait around for the I/O to complete before it can go on with its other processing. It is the difference between waiting 20 minutes at Piz-

za Hut for your pizza to be made and calling ahead so that the only time you spend is on the phone to place your order and walking in to pick it up.

Originally we believed that Ada tasking would be the key to the no-wait I/O requirement. When we tried to implement it, tasking was slow. While a particular task waited for a rendezvous, the entire os_process would come to a halt.

A system of interfaces was devised to serve as a communication block between I/O-requesting and I/O-processing software. The interface structure, made up of an interface record array and buffers, serves as a queue that requests can be immediately stored in so the user can continue processing. When the I/O-processing (I/OP) has time to execute, it reads a request off the interface record array, reads the associated data from the appropriate buffers, and completes the request. The interface structure holds all information that is needed to process the request and the request's status.

b. <u>I/O Across Processors</u>

The RTSE is required to support I/O regardless of what processor the requesting code is running on. The system's solution to APU I/O would cause the APU tasks to execute in an unpredictable fashion.

Our solution lies is in the use of the interface structure. All I/O requests are placed in an interface record regardless of where the task is running. Specifically, an APU task's I/O request is placed in an interface record and the task continues without influencing its execution order. The I/O processing software always runs on a CPU where execution order is strictly maintained.

c. <u>I/O Across Clusters</u>

While each cluster has its own disk, the customer required that disk writes occur on only one disk when running integrated. Our O/S / Ada does not support I/O across clusters. RTSE solved this dilemma using a system of slave clusters and a single master cluster. Each cluster, whether master or slave, has its own interface structure, I/O Processor, and disk. Recall that a cluster is made up of a CPU and one or more APUs (Figure 3). The clusters designated as slave have an I/O Processor to execute all requests stored in the local area of their interface record (Figure 8). Requests in the local area are performed on the disk associated with that cluster. The master cluster has access to the master portion of all the interface record arrays. By storing all output requests when integrated in the master portion of the interface record array, all write requests will be performed on the master disk when integrated, just as required.

d. <u>Disk I/O : direct_io</u>

An effort was made to make the I/O requesting software structures and formats Ada-like; this was done to provide the user with a familiar format. The result was the generic Request_direct_io package. The no-wait I/O request software mimics direct_io with similar procedure and parameter names being used. When the request is processed by the "I/O Processor" software, it uses Ada's direct_io, making this service software transportable to other hardware/compiler combinations.

e. <u>I/O for Objects Greater than 64 K</u>

Since request_direct_io is an extension of direct_io for the RTSE, it carries with it direct_io implementation limitations. While in the preliminary design phase, we considered breaking up the data into sizes direct_io could handle but there was an unacceptable cost attached in the form of overhead time and space, and lost Data Abstraction.

Since machine-specific direct_io object size limits already removed an element of portability and standard Ada constructs were not suitable for the situation, software was developed that interfaced with the operating system services. Contiguous_io was the result.

To the user the choice between request_direct_io or request_contiguous_io is based on the size of the object. While contiguous I/O allows faster retrieval due to its file structure, it is used primarily because it allows unlimited object size and only one stable copy of the object exists at any one time. Direct_io's implementation keeps one or more copies of the object in system data buffers and multiple copies in an array of user buffers. As the impact of this has systemwide effects, guidelines based on the size of the objects were establish to indicate the type of I/O to use.

f. <u>Output Messages to Error Logs, Audit Logs, and Terminal in Specific Format</u>

Besides two-way communication with files, there was a need to for a write-only utility. This utility is strictly for the output of strings, hence need not be a generic. Any Ada object can be translated into a string by use of the image attribute.[2] The message_request procedure for the user and I/O Processor's message_request operations compose this implementation.

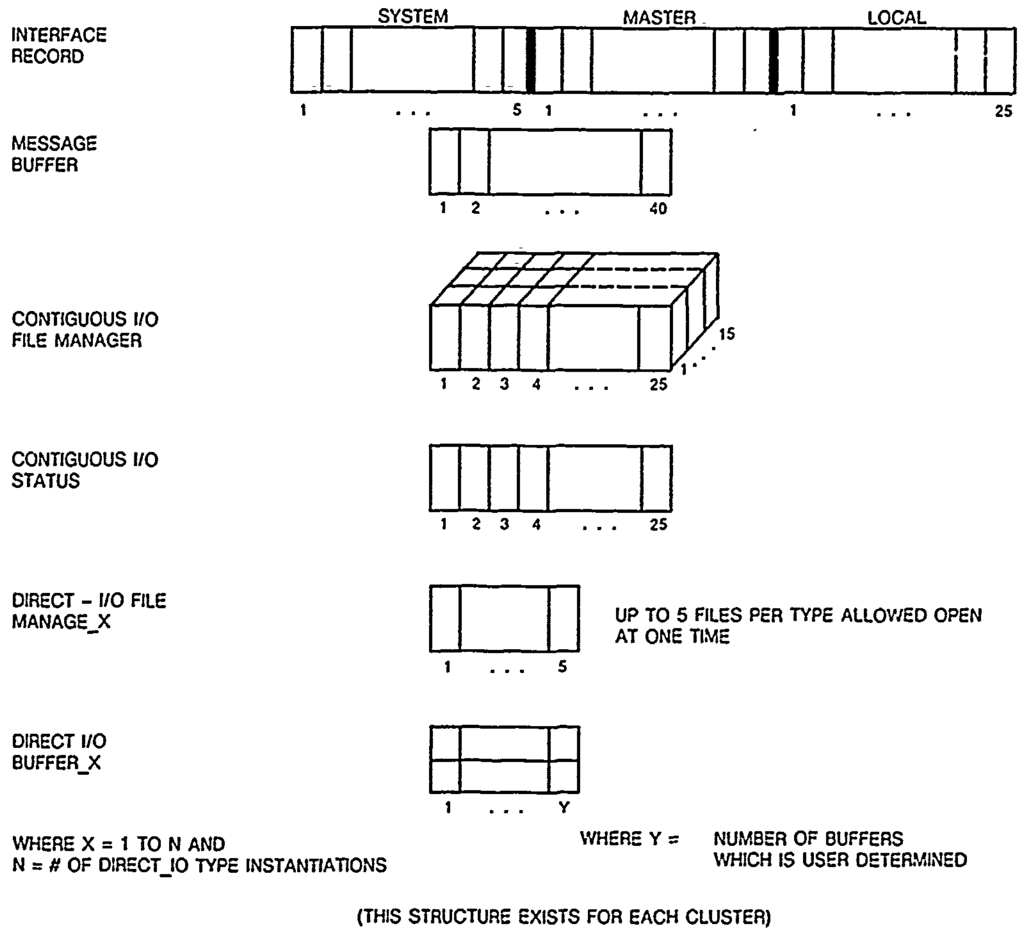


Figure 8 I/O : Interface Structure

g. Disk Marking

Since the O/S did not provide the user a service to request and status disk marking, that too became a RTSE I/O requirement. Portability was not possible given the fact that our Ada version did not provide an interface and there were no O/S services to indicate or modify the status of a disk available to the user. The devised interface used a pragma interface to assembly code: a machine-dependent implementation.

h. Additional Software Engineering Requirements

To make the software portable for use on other hardware or with other compilers, we made an effort to use the standard Ada constructs of text_io and direct_io while avoiding system-dependent tools. Reusability was achieved through creation of generics which were instantiated mul tiple times within the simulation software and perhaps can be used on future projects. Generics and packaging added to the maintainability by limiting the scope of changes.

Another way to improve maintainability is to increase software readability. We encouraged the use of enumeration types. We implemented an English-like naming practice for types, objects, and packages. We required that acronyms and abbreviations be defined in the prologue of each unit in which they appear.

The quest for data abstraction influenced the allocation of objects and functions into packages and the grouping of data into records. For contiguous_io we provided a procedure which provides the exact memory location of an object rather than allowing the user to calculate this information. Good data abstraction is attained by not allowing options to go around this structured method. By forcing the user of contiguous_io to use this function to determine the address of an object in memory, a degree of control and a bit of calculation is automated, removing a location for potential user errors. An error of sending the wrong address could be fatal to the executing software if an address for the executable code rather than the desired object is written over. Re-

moving this possibility can only be a boon to the I/O system.

Implementation - The no-wait adaptation of I/O was coined Real-Time I/O (RTIO). To meet the requirements, the RTSE was to supply the user with four I/O related abilities: no-wait disk I/O, no-wait disk I/O for types greater than 64 K in size, message I/O, and disk marking. The user refers to these services by the following names: request_direct_io, request_contiguous_io, message_request, and disk_marking. Collectively, we refer to the software the user interfaces with as the "User Module". The User Module software fills the appropriate interface structure arrays and buffers.

For example, when a user requests a message, the message is stored in an available spot in the message_buffer and an interface record is set up with the corresponding message_buffer index and type of message request stored. Pending is stored as the status in the interface record. The user who made the request can then continue processing while the request waits in the interface record queue for the I/OP.

The interface structure is used in conjunction with all I/O requests, providing a separate buffer segment in the interface_record for system, master, and local information (Figure 8). Additionally there is one Message_buffer, one Contiguous_io_file_manager, and one Contiguous_io_status buffer per cluster. Conversely, there exists one direct_io_file_manager and one direct_io_buffer for each type instantiation of direct_io. This number directly correlates to the number of request_direct_io instantiations and is user determined.

The key to the interface between the User Module software and the I/OP software is the interface record, which is made up of three parts: system, master, and local. Each is a circular fill queue. The system interface record is reserved to report internal RTIO errors. I/O requests to the master cluster are queued in the master section,and I/O requests for the local I/O Processor are queued in the local portion.

The I/OP functioning is based on how we had hoped Ada tasking would work. A simplified structure follows. The I/OP consists of a big loop which steps through the interface record array checking the statuses. Using a case statement which keys off the status, and another which keys off the type of request when the status is pending, the I/OP executes the desired I/O requests as appropriate. As an I/OP is on the CPU processor of each cluster, it can be linked at an appropriate priority so as to not interfere with the execution of the other tasks, hence run invisibly to the simulation software.

To circumvent the limitation of excluding packages as allowable formal generic parameters, the principle of the pseudo-generic was used for the I/O Processor (I/OP). I/OP software refers to a predefined set of interface arrays, objects, and procedures while each cluster and user determine their own unique names. This pseudo-generic principle comes into effect as a series of renames causes the unique packages to be renamed to the predefined set that the I/OP software recognizes.

SUMMARY

The development of the Real-Time Simulation Environment for the B-2 ATD was a challenging undertaking even though many other environments had been developed before. Many felt that the challenges encountered were a direct result of this being an Ada project. Although the lack of experience in Ada on an actual simulation project was a contributing factor, we have judged the following factors to be of greater influence in meeting our challenges.

The "early adopters syndrome"[1] most likely had the greatest impact on our development effort. This syndrome is characterized by the availability of only a few validated Ada compilers and even fewer mature Ada compilers. Our compiler was immature at the start of the program and has gone through rapid changes and multiple revisions. We too have had to adapt to these changes, report unfavorable situations not detected by the validation tests, and accept the workarounds provided until the next revision was available. While we believe that these inconveniences were common to most compiler vendor/customer teams at the time, they are unique to immature compilers.

Some unique Ada features were handled differently than were originally anticipated. The Ada tasking model provided had significantly impacted the design of the control and I/O sections of the RTSE. If the essence of Ada tasking can be accomplished via an interface to O/S services, then it is possible for a compiler to do the same or better. The compiler that more fully exploits the intent of Ada tasking and functions in a reasonable amount of time will have a significant advantage over its competitors in the simulation market.

The size of the B-2 ATD had a significant role in determining the direction of the development process. However, we believe that the structure inherent in Ada had a positive influence on the development process. This is especially true now as the project and compilers are maturing. This view is

consistent with the life cycle cost studies done on Ada and other languages.

The failure of Ada to support the passing of packages as formal generic parameters had a negative impact on the development of the control and I/O sections of the RTSE. Several legitimate ways were found to approximate the effect. As mentioned earlier, the methcds were package renaming and the use of built software from standardized templates. However, the use of packages as formal generic parameters would have made this task easier and more maintainable.

Since Ada does not recognize multiple os_processes, many schemes have been developed to provide communication between os_processes. Although our experiences do not include an exhaustive list, the method provided by our current vendor is by far the best system we have seen. This method provides for the linking of Ada packages into a common identifiable area. This area can then be linked against when linking os_processes. These os_processes can then utilize the data in those packages. Other systems we have investigated only recognize the sharing of individual objects between os_processes. Even with renaming of objects, it can be considerably more difficult to map packages through discrete objects. It is our opinion that compiler vendors who want to compete for large multiprocess Ada projects will have to support inter-process communication at the Ada package level.

We feel that the advantages of an Ada-based system greatly outweighed the challenges imposed by Ada and the related "early adopters syndrome". As we look forward to new projects and possible rehosting on different hardware, we have already observed benefits of Ada.

REFERENCES

1. "Ada Issues Awareness Seminar" by Reifer Consultants, Inc., 25550 Hawthorne Boulevard, Suite 208/Torrance, CA 90505

2. ANSI/MIL-STD-1815A - 1983, "Ada Programming Language"

ABOUT THE AUTHORS

Walter E. Zink is currently a Section Head – System Engineering at CAE-Link Corporation, where he has held management and technical positions since 1986. His current effort is devoted primarily to system-level software design of the real-time environment that supports the Ada architecture in use at Link. Prior to that, Mr. Zink was employed by the General Electric Company for twelve years, where he held various technical and management positions. His principal activities there were in the areas of Computer Based Instruction and Artificial Intelligence/Expert Systems. Mr. Zink holds a B.S. degree in physics from Harding University.

Jill Neebe is a Systems Engineer at CAE-Link Corporation, where she has been employed since 1987. Her current effort is devoted primarily to system-level testing of real-time Ada software. Prior to this, Ms. Neebe worked in Software Quality Assurance developing and implementing instructions for quality assurance in a DOD-STD-2167A environment. Ms. Neebe holds a B.S. degree in mathematics from Virginia Polytechnic Institute and State University.

DRLMS TECHNOLOGY - A CRITICAL ASSESSMENT OF THE STATE-OF-THE-ART

John D. Stengel, Jr.
Science Applications International Corporation
Dayton, Ohio

Thomas W. Hoog
United States Air Force/Aeronautical Systems Division
Wright-Patterson AFB, Ohio

ABSTRACT

It was 20 years ago that the training community stopped to make an assessment of radar simulation fidelity and effectiveness. An evaluation of analog systems, which had served well for many years, identified that many of the inherent limitations could be overcome if modern digital technology were applied. Project 1183 and the acquisition of digital radar landmass simulation (DRLMS) systems for the Navy A-6E Weapon System Trainer and Air Force Undergraduate Navigation Training System helped make the transition from analog to digital radar training systems a reality. During the last 10 years, radar simulation technology has been significantly impacted with the introduction of training requirements for high resolution synthetic aperture radar (SAR) systems. Now, in 1991, we are again at a time appropriate to reevaluate the progress we have made and the effectiveness of today's DRLMS systems. The objective of this paper will be to provide a brief history of radar simulation, make an assessment of the successes as well as specific problems and issues associated with the simulation of high resolution radar systems, identify an approach based on video processing of optical sources that could lead to satisfying many current and future radar simulation requirements, and introduce alternative approaches for specifying the performance of future DRLMS systems based on a more rigorous assessment of training needs and the benefits that might be anticipated.

INTRODUCTION

Digital Radar Landmass Simulation (DRLMS) has come a long way during its 20 year life. Some bold decisions were needed and made along the way that brought us to where we are today with highly realistic radar displays capable of supporting many types of training. However, limitations do exist that cannot be economically overcome by just advancing technology again. Some tough decisions need to be made that will optimize the tradeoff to be made resulting in operational training needs being met utilizing all the relevant technology available and within a reasonable cost.

HISTORY

Project 1183

In the early 1970's, ground mapping radar simulation technology took a significant step with the use of digital processing of data to create the simulated radar video. These simulations utilized the same source data as the analog systems except in a digitized form. In 1973 a major step was taken by starting an R&D program, Project 1183, that employed new concepts in both database and processing technology.[1] This program addressed the following limitations of the existing F-111 simulator:

- Inadequate breakups of cultural features
- Lack of height data for cultural features
- Costly and lengthy process to make even minor updates to the transparency and database
- Unusable display on the attack radar on the short range scales
- Unrealistic simulation of the terrain following radar system
- Maintenance problems due to the complex analog servo mechanism and analog signal processor.
- Inability to consistently reset the simulation to precise geographic positions and obtain precise repetition of the simulation over a previously flown ground track.
- Point features aligned in the cardinal direction instead of their true orientation.

Project 1183 and the Navy's A-6E DRLMS program, which started a little later, demonstrated that all these limitations could be eliminated or significantly improved.[2]

Radar Data Base Evolution

One of the key reasons for the very significant improvement in ground mapping radar simulation was the completely new approach to the source database. The Defense Mapping Agency (DMA) produced a multi-level/multi-resolution digital database that described both terrain elevation and the features that sat on top of the terrain. These features were described using physical characteristics (material type, feature identification, definitive outline of feature, etc.). Because these database descriptions were generic, the DMA product could support any radar simulation through the development of a transformation program.[3] This was the foundation of this new concept defined by ASD and was essential to the development of an economical approach to digital radar landmass simulation. In the late 1970's many new simulator programs adopted this proven concept, and the need for extremely large quantities of digital data from DMA was identified. During the middle and late 1970's DMA established multinational agreements to produce data. This increased the production capacity for digital data and resulted in a further expansion of the database support concept for radar simulation with the British and German Tornado simulators.

Standardization Of Requirements

In the late 1970's, when several new Air Force simulator procurements were being planned that included DRLMS systems, a small group of Air Force engineers was tasked to produce a generic set of DRLMS requirements. The resulting document was the product of much soul searching and incorporated the lessons learned from all previous DRLMS developments and products from the three DRLMS vendors at the time.[4] This document was provided to these vendors for comment. The results of these activities were:

- A quantitative requirement that defined the DRLMS processing and memory capacities based on a DMA data density curve and feature location accuracy.
- A quantitative requirement that defined the fidelity of terrain reconstruction techniques based on the roughness of the terrain and the range scale selected for display.
- Identification of specific radar effects significant to training.
- A requirement to provide specific test features to aid in the test and evaluation of the display as well as a diagnostic tool.
- The requirement to provide a means to make simple modifications to the DRLMS database to support the insertion of mission specific features and/or the correction of minor database er[illegible]

As DRLMS technology matured along with the DMA database production methods, some new concerns came into prominence including digital processing of radar signals in the airborne radars, the cost of producing the DMA digital database in the resolution levels and in the quantity desired by the operational commands, and the need to determine how much fidelity is enough to satisfy operational training requirements. Newly defined requirements for high resolution radar simulation were based on engineering analyses of operator performance rather than as a result of controlled research using operational scenarios. However, because of the large apparent increase in feature content on the high resolution radar display, issues associated with source data bases again became critical.

Transition To Synthetic Aperture Radar Requirements

By the late 70's, synthetic aperture radar (SAR) systems were being developed in the laboratories that could produce extremely high resolution displays. Samples of Forward Looking Multi-Mode Radar (FLAMR) data, an advanced development program, were provided to simulator developers and DMA to assess how this type of radar would impact DRLMS technology. The initial consensus from industry was that the database would need improvement but that processing technology was basically adequate even with the need to simulate SAR unique effects. One of the first programs to address the requirements associated with high resolution radar simulation was the F-16 DRLMS system development program which featured a doppler beam sharpened mode.

The Analytic Sciences Corporation (TASC), under contract to the Air Force, Aeronautical Systems Division (ASD), accomplished an analysis in 1983 to illustrate the characteristics of SAR imagery which should be incorporated into a

DRLMS system and to then evaluate the suitability of existing DRLMS technology for simulating these characteristics.[5] This analysis was soon followed in 1984 by a similar study by Link Flight Simulation Division, under contract to the Navy, Naval Training Equipment Center (NTEC), to explore the implications of Doppler beam sharpened radar and synthetic aperture radar in training requirements.[6]

Conclusions reached by both analyses were similar in nature. First, it was noted that although the state-of-the-art in digital technology was adequate to support SAR simulation, existing DRLMS systems would require significant redesign to successfully meet training requirements. Second, a series of specific radar effects was identified as being uniquely characteristic of a SAR map. SAR unique effects identified by both reports included scintillation, motion compensation errors, moving targets, range/range-rate mapping distortions, feature layover, azimuth/range sidelobe overloading, and range foreshortening. However, of potentially greatest significance, both reports identified the need for a more detailed, higher resolution source data base as a basic necessity for realistic SAR simulation. The Link report specifically identified the need for a simulation data base equivalent in terms of feature content and resolution to DMA's Level X prototype data base.

These additional phenomena and the lack of operational experience with SAR systems increased the dilemma of how much simulation fidelity was needed to satisfactorily train radar operators. Up to this point, the primary performance requirements were based on engineering analyses of the ability of a radar operator to finely tune his scope and lay cursors on an aim point. This analytical determination of requirements was satisfactory, but begged the issue of experimentally derived data to demonstrate whether this approach resulted in over specification or under specification of performance requirements. The Air Force laboratories were challenged with the difficult problem of determining how accurate feature location had to be maintained, to what fidelity feature shapes had to be maintained given all the information available to an aircrew. Digital processing of radar video further compounded this issue and added some additional questions due to characteristics of the processor.

It was apparent that the content of the source data base had to be addressed. It also became obvious that the cost of database production was becoming extremely large. One of the first efforts to address this problem was to enhance the existing database artificially. The feasibility was demonstrated in 1978 and again in the early 1980s by generating synthetic features in large DMA database homogeneous features (residential areas, factory complexes).[7] This was done without any additional descriptors required in the DMA database. From the outset it was realized that this approach could not be applied to key radar significant features, but was viable for large non-radar significant homogeneous features. Still, there were skeptics among the user community and some database purists.

It was suggested to industry that they apply the texture schemes used in visual simulation to radar simulation. Some of the initial texture patterns were very artificial; however, they matured rapidly. This sharing of development history helped to bring the visual and radar simulation communities even closer. This approach was successfully used on the F-16 DRLMS and subsequently on the F-15E, B-1B, and B-2.

ASSESSMENT OF CURRENT CAPABILITY

DRLMS System Performance

The simulation industry has made tremendous strides developing the state-of-the-art in DRLMS technology over the last 20 years. The quality of simulated imagery, particularly for real beam systems, has improved to the point where target analysis and radar predictions can be accomplished with the support of a DRLMS system. Problems associated with quantization effects (e.g., number of reflectance codes, number of video output levels, antenna beam implementation, etc.) from various stages of the digital processing and data storage observable on the radar display have largely been overcome. Fundamental radar effects such as terrain shadowing, aspect geometry, and directionality effects are well understood and effectively modeled in most DRLMS products. Creativity by DRLMS vendors such as GE, Link, Boeing, Loral, Harris, and Merit Technology have resulted in innovative ways of providing enhancements (i.e., texture, artificial feature implementation, and multi-level data base merging) to the simulated image that result in improved fidelity. Technology growth has had a significant impact on improving system reliability,

maintainability, and availability while at the same time reducing both development and life-cycle costs.

DMA Data Base Concept

A major contributor to the successful evolution of DRLMS systems has been the application of the multi-level DMA digital database.[8] Standard DMA digital products developed to support radar simulation have included Levels 1 and 2 DFAD. Both products were originally intended to be used as source data for real beam radar simulation - Level 1 for large area coverage for general navigation and Level 2 for increased detail surrounding radar fix points (RFPs) and offset aimpoints (OAPs).

The database transformation concept has proven successful and has permitted a single source database (DMA) to be tailored to a specific radar simulation. DMA is currently supporting radar data base transformation programs for the A-6E, E-2C, EA-6B, B-52, B-1B, C-130, EF-111A, and F-16 WSTs.[9] An inherent capability associated with the implementation of digital systems that has been exploited by DRLMS vendors is the data base update capability that permits the end user to easily make simple modifications to the radar database through the use of an update console.

Although problems associated with the use of DMA data were initially identified (e.g., digitizing anomalies, differences in how DMA analysts interpreted and encoded source data, areas with sparse or missing data, etc.), the overall quality and availability of the DMA products has steadily improved. Further improvements can be expected as DMA progresses with the implementation of their Digital Production System scheduled for completion in 1992. This new system, referred to as DPS MARK 90, will provide an all-digital production system for increased throughput, greater product flexibility, and improved responsiveness.

High Resolution Radar Simulation Data Bases

A fundamental aspect of DRLMS system requirements and design approaches is that the DMA data, at whatever level available for a given application, is the ground truth representation for simulation purposes. The fidelity of the DRLMS ground truth image is, therefore, only as good as the DMA representation from both a feature portrayal and descriptive information perspective. Although a DRLMS vendor can provide enhancements to the simulated imagery to improve the qualitative realism, correlation to the Earth's surface will be no better than what is described in the data base product.

Analyses of high resolution SAR imagery for weapon systems such as the B-1B and F-15E indicated that the original DMA products Levels 1 and 2 would not suffice as the only source for generating simulated images. Based upon system resolution performance of better than 10 feet, specific data base limitations that were evident included a lack of continuous lines of communications, inadequate feature content and detail, and a general lack of adequate descriptive information.

A joint ASD/DMA data base requirements definition process was initiated in 1984 to specifically identify the content and format for a new product intended to support the simulation of high resolution radar systems. The outcome of this effort was a draft specification and prototype DFAD product referred to as Level X.[10] Level X data contained significantly more scene content and detail than any other existing DMA product and was met by much enthusiasm by the DoD training community. Level X data was evaluated by many DRLMS vendors and became the preferred solution for meeting the B-1B WST HRGM simulation requirements. However, after producing a limited number of Level X areas for the B-1B, DMA determined that the cost of turning Level X into a standard product would be prohibitively high and further efforts were discontinued.

Further high resolution data base requirements analyses conducted by ASD with the support of the Air Force Human Resources Laboratory (AFHRL) and Armstrong Aerospace Medical Research Laboratory (AAMRL) after the discontinuance of Level X resulted in the definition of two new products that could be more economically produced - Level 3C (compiled from 1:50,000 scale map source) and Level 2E (based on the existing Level 2 with more complete lines of communication representation).[11] Based on user requirements, specific RFP and OAP information can be provided in both Levels 3C and 2E. It is currently envisioned by DMA that DFAD Level 2E and Level 3C will remain the standard products to support high resolution radar simulation requirements for at least the coming decade.

The current assessment of Levels 2E and 3C is that for relatively isolated RFPs and OAPs, the basic information including the specific feature of interest and prominent surrounding features will be

provided. However, the challenge of transforming these products into a DRLMS on-line data base capable of supporting high resolution radar simulation with a high degree of image fidelity will remain with the system developers. Information relative to the precise nature of feature detail, terrain characteristics, and foliage appearance will require the DRLMS system developer to infer as much as possible from the data base descriptors and apply artistic license to the degree desired in order to produce simulated imagery with similar attributes to that of the actual system.

High Resolution Radar Effects

The physics associated with fundamental radar effects such as terrain shadowing, aspect geometry, and directionality effects are well understood and effectively modeled by most DRLMS vendors. These radar effects provide the basic character of real beam imagery and can be justified from a training requirements perspective. Analyses conducted of high resolution systems have also provided a comprehensive definition of those characteristics unique to digital processing of doppler phase histories, and appropriate models have been implemented in existing systems. What is not clear, however, is the degree of fidelity afforded the simulated imagery by implementing a rigorous set of SAR algorithms and the cost from both a development and system complexity perspective. The bottom line is that the training utility of these simulated effects relative to the added system complexity and cost has not been well established.

ALTERNATIVE APPROACH

Recent efforts have resulted in significant improvements to the overall fidelity of the simulated high resolution imagery. However, each of these approaches is dependent upon information contained in the DMA DFAD product and associated enhancements resulting from DFAD feature descriptors (e.g., surface material, feature identification, etc.). Simulation results compared to an actual radar presentation are limited by the information contained in the DMA product.

This same problem was encountered when trying to utilize DMA DFAD products as the basis for visual data bases to support computer generated simulation imagery for visual systems. The real time application of different synthetic texture patterns based on DFAD feature descriptors has been successfully utilized with simulator visual systems for a number of years. However, of greater significance is the application of overhead imagery as a source for geographically specific photographic based texture. [12,13,14]

Science Applications International Corporation (SAIC) has developed and demonstrated a similar approach for radar simulation applications. Between January 1984 - January 1986, SAIC's Aeronautical System Operation in Dayton, Ohio completed the development of two B-1B Engineering Research Simulators (ERS) for the Armstrong Aerospace Medical Research Laboratory (AAMRL) and Strategic Air Command (SAC). The B-1B ERS is a real-time, man in the loop simulator designed to model and emulate the physical appearance and functional characteristics of the B-1B strategic bombers' flight and aft crewstations. Of prime concern to the B-1B Offensive System Operators (OSO) is a realistic simulation of the B-1B's High Resolution Ground Map (HRGM) synthetic aperture radar system.

The selected radar simulation approach featured a custom electronics module based on conventional DRLMS design which could provide real-time simulation of the B-1B's real beam radar mode. The High Resolution Ground Map (HRGM) mode simulation approach, however, was based on preprocessed fixpoint imagery generated off-line and stored on laser disk for data retrieval during training. While several data base sources (including high altitude photographs and actual SAR imagery) were originally evaluated, DFAD Level 1 was selected for production of the B-1B ERS's HRGM imagery because of coverage, cost, and schedule considerations. HRGM data bases are stored in a gridded format containing gray scales/reflectance values which represent the predesignated radar fix point (RFP) areas in two dimensions. Multiple data bases, each with a grid size corresponding to one of the available radar map resolutions, are developed for each RFP. Each data base map has a single set of geographic coordinates referenced at the map center that are used for display computations relative to the aircraft location. This approach captured some of the geographical content of the fixpoint area needed for procedural training; however, the resulting imagery was simplistic in appearance and did not contain the scene content observed with actual HRGM imagery.

In support of human engineering research for the Armstrong Aerospace Medical Research

Laboratory (AAMRL) and the B-1B SPO, SAIC was tasked to improve the fidelity of the B-1B ERS simulated HRGM imagery. After completing a review of alternative sources, it was found that United States Geological Survey (USGS) high altitude photographic data was available for most regions in the continental United States including SAC's Strategic Training Route Complex (STRC). Using this USGS photographic data as source material, and a variety of commercially available hardware and software products, a data base transformation process capable of supporting simulated HRGM images was developed.

The process developed to generate on-line HRGM data bases and provide real-time processing is illustrated in Figure 1 and includes the acquisition of source imagery, digitization, transformation of imagery and terrain elevation data into a video format for real-time processing, and storage on video disk. The transformation process requires that the data base photography be analyzed to determine radar unique feature characteristics which are subsequently assigned an appropriate gray scale/reflectance value. Particular attention is given to buildings and significant man made structures, land/water contrasts, no-show areas such as roads and runways, and vegetation. Surface texture is also added to enhance the overall image appearance. DMA Level 1 DTED is used as the terrain elevation source for computing occulting effects both within and to the selected map area. As part of the transformation process, a terrain elevation value is computed by interpolating between 3 arc second DMA Level 1 DTED values and storing each corresponding grid element in the feature map. Real-time data retrieval of the digitized radar imagery is accomplished on a scan by scan basis by the DRLMS. Each scan line contains approximately 400 range elements to approximate the required resolution. Grid cell gray scale values retrieved by individual scan lines are then mapped into the DRLMS display memory. Figure 2 illustrates a high altitude photo for a "typical" area of interest, and Figure 3 a simulated HRGM image produced using the described process.

In addition to the data obtained from USGS, source data was also obtained from LandSat (Thematic Mapper System). A significant advantage is that the LandSat data may be obtained in a digital format thereby eliminating one of the data base generation steps. However, the resolution of this data was less than that available from USGS and would result in a lower fidelity simulated image. SPOT imagery is another candidate data base source but was unavailable during the evaluation period.

The photobased approach provides a number of significant improvements for high resolution radar simulation that include enhanced image resolution and scene content that are highly representative of actual radar system imagery. The current process faithfully retains all characteristics of both the source imagery and the terrain elevation data. Information

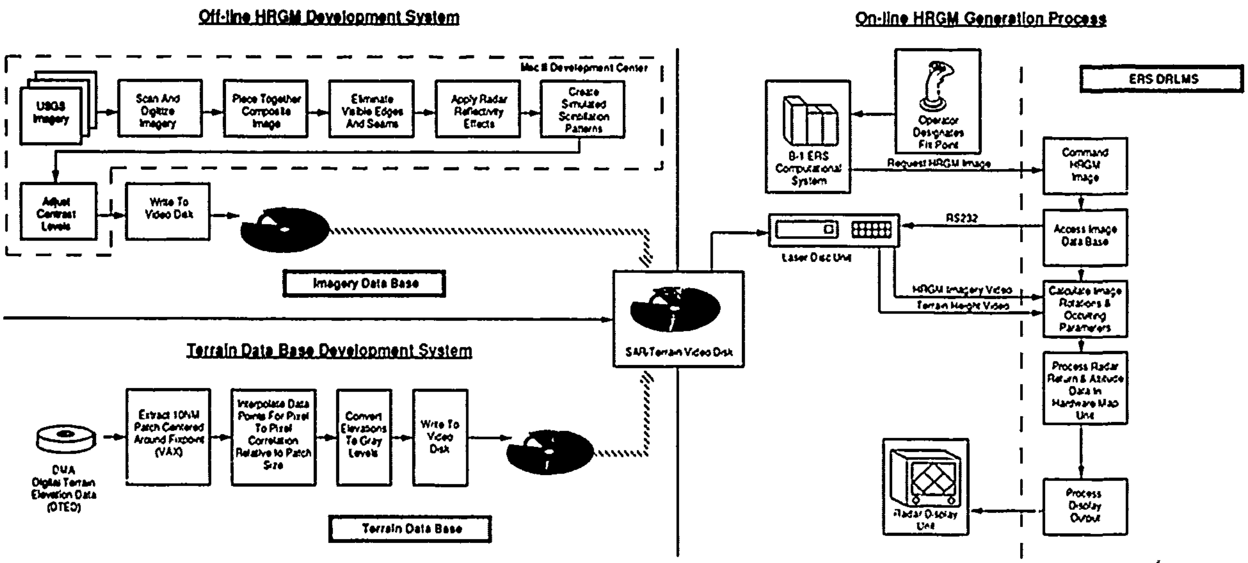


FIGURE 1 HRGM DATA BASE AND REAL-TIME PROCESSING

FIGURE 2 HIGH ALTITUDE PHOTOGRAPH OF AREA OF INTEREST

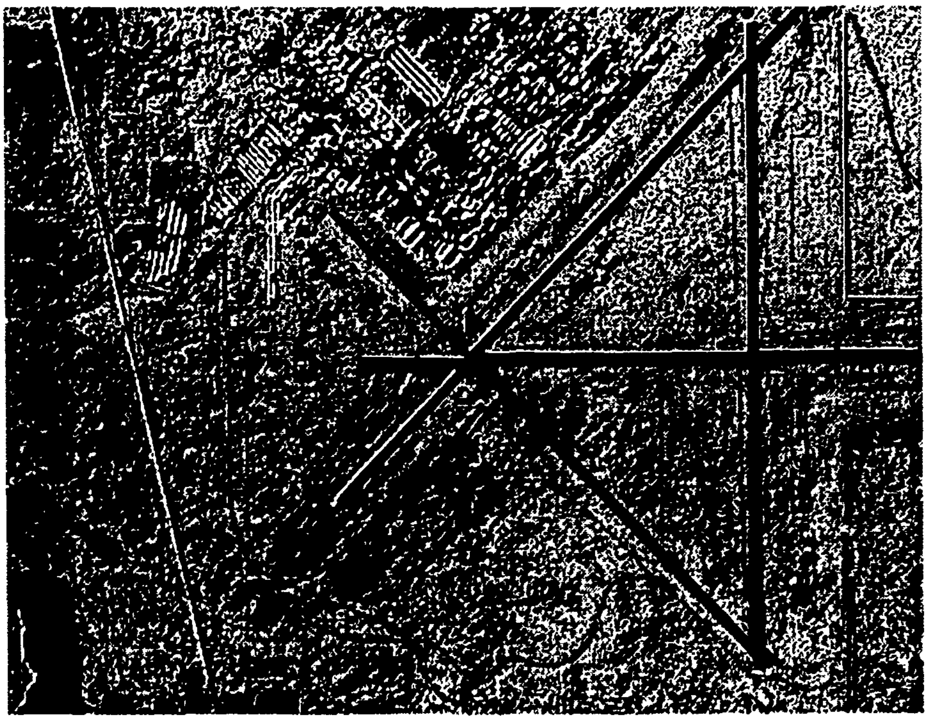


FIGURE 3 SIMULATED HRGM IMAGERY

collected from interviews with B-1B OSOs located at Ellsworth AFB who evaluated the simulated B-1B HRGM imagery indicated that it compared favorably to the actual aircraft radar system imagery. Although the transformation process and subsequent real time processing result in simulated image characteristics (e.g., feature intensities, terrain occulting, land/water contrast, etc.) that are representative of the actual aircraft radar system, further development in the areas of cultural occulting, the simulation of specific geometrical effects such as horizontal/vertical aspect, and implementation additional SAR unique characteristics needs to be accomplished.

CONCLUSION

As was related earlier, the principle limiting factor in large area, high fidelity DRLMS systems is the source data base. Given this fact, one needs to question the value of high fidelity modeling efforts of both the radar system and the phenomena due to geometry and physical laws if an adequate data base is not available. This is not to imply that high performance ground mapping radar simulation is not required but whether lower cost simulations with good performance will satisfy many training needs without some of the "special effects" due to doppler processing, seldom encountered geometry, extreme antenna pointing angle situations, or unusual operator use conditions. The following is suggested during the requirements definition phase of a program to determine the performance fidelity needed and the acquisition strategy to be followed.

The specific radar skills that need to be trained should be carefully evaluated to determine the most cost effective method of instruction. Careful consideration needs to be given to whether the objective of the DRLMS system is procedural in nature or whether advanced skills requiring target identification and radar scope interpretation (RSI) need to be trained. In the past, many Air Force radar navigators/weapon system operators have expressed the belief that the training benefits associated with radar simulation become less significant after a student has had several actual flights in the aircraft. On the other hand, the perishability of RSI skills also needs to be considered. If there are an insufficient number of aircraft training flights necessary to permit the needed skills to be retained, then ground based training must fill the void.

Since high fidelity DRLMS systems with large, high fidelity data bases are expensive, alternatives to these systems as an integral part of full mission simulators or weapon system trainers need to be examined. Procedural training might be achieved using a lower performance DRLMS system that is fully integrated with the weapon system trainer. Requirements related to system processing performance (e.g., accuracy, data density, etc.) and radar effects fidelity for DRLMS systems supporting procedural training would need to be reassessed.

A limited number of high performance DRLMS part task systems might then be developed as part of the suite of training devices for the weapon system. These part task trainers could be devoted to training perishable skills that require high performance/high cost capabilities such as those associated with target identification, target prediction, and RSI. A high fidelity part task trainer would not require the complex interfaces with navigation or weapon delivery subsystems, nor would they require large, high fidelity source databases. Alternative DRLMS architectures or hybrid systems utilizing a photobased data base enhancement technique like that described earlier might be exploited to meet the part task trainer high fidelity simulation requirements. Revisions to existing DRLMS performance specifications would need to be addressed if the application of photo texturing techniques is to be considered.

Technology is available to support high fidelity real time ground mapping radar simulation. However, the database to support this level of fidelity is not available in sufficient quantities to meet operational training needs and the cost of producing such data is currently prohibitive. Enhancements are possible that contribute to realism and should contribute to training utility. But, some difficult decisions and smart choices must be made. In order to achieve the proper mix of radar training media suggested earlier, the end user and acquisition organization must conduct the appropriate trade studies. The specific tasks and a detailed analysis of how these tasks are accomplished must be accomplished in order to reach the most cost effective solution. Trade studies must also be accomplished to develop a more definitive understanding of mission rehearsal training requirements.

Performance compromises are often possible, but they must be made in the right areas. It is also essential that the contractor be completely aware of the results of these trade-offs so that the training philosophy is carried into the design implementation phase of the program. The entire team - end user, acquisition organization, and contractor - must seek to maximize performance for each new application by taking advantage of simple database enhancements, by developing alternative data base texture capabilities, and by selecting the appropriate mix of training devices (including simplified radar simulation systems) whenever the opportunity is presented.

REFERENCES

1. Hoog, T.W., "Project 1183 - Digital Radar Landmass Simulation (DRLMS) Development ," 7th NTEC Proceedings, 19-21 Nov 1974, pp. 55-79.

2. Air Force Project 1183 Digital Radar Landmass Simulation (DRLMS) Development Test And Evaluation (DT&E) Report, 25 July 1978.

3. ASD Exhibit 75-5, General Exhibit For A Digital Data Base Transformation Computer Program, Nov 1975.

4. ASD Exhibit 75-7, General Exhibit For A Digital Radar Landmass Simulator (DRLMS) System, Nov 1975.

5. Avila, P.G., Keydel, E.R., Kronenfeld, J.E., and Pinto, R.W., "Real-Time SAR Simulation Analysis", The Analytic Sciences Corporation, Technical Report No. TR-4089-2, 1 April 1983.

6. Peters, R.L., "Synthetic Aperture Radar Simulation Study", Link Flight Simulation Division, Singer Co., Technical Report No. NAVTRAEQUIPCEN 83-C-0068, March 1984.

7. Dal Sasso, A.J., Dwyer, T.M., and Kalinyak, R.G., "Synthetic Enhancement Of Defense Mapping Agency Data", NAECON '84 Proceedings, 21-25 May 1984, pp. 1057-1063.

8. Defense Mapping Agency. Product Specifications for Digital Feature Analysis Data (DFAD) Data Base Level 1 and Level 2, Second Edition, 1986.

9. Defense Mapping Agency, " Digitizing The Future", Third Edition, DMA Stock No. DDIPDIGITALPAC

10. Defense Mapping Agency, Product Specification Supporting Prototype Level X For STRC , First Edition (Draft), November 1984.

11. Bell, H.H. and Crane, P.M., "Flight Training Simulators: Evaluation Of DMA Level 3C Feature Analysis Data For Simulating Synthetic Aperture Radar", Air Force Human Resources Laboratory, Operations Training Division, September 1988.

12. Economy, R., Ellis, J. R., and Ferguson, R. L., "The Application Of Aerial Photography and Satellite Imagery To Flight Simulation," 10th IITEC Proceedings, Nov 29 - Dec 1 1988, pp. 280-287.

13. Brown, T. and Wilkerson, T.C., "Scene Realism: The Synergy Of Data Base Technology And CIG Hardware," 10th IITEC Proceedings, Nov 29 - Dec 1 1988, pp. 288-294.

14. Kraemer, W. and Moberg, T., "Visual Simulation Utilizing Computer-Reconstructed Images From Scene Photographs," 10th IITEC Proceedings, Nov 29 - Dec 1 1988, pp. 295-304.

ABOUT THE AUTHORS

John D. Stengel is the Manager of the Real Time Systems Division at the Aeronautical Systems Operation of Science Applications International Corporation (SAIC). Prior to joining SAIC, Mr. Stengel held a variety of positions within the United States Air Force, Aeronautical Systems Division, in support of flight simulation and training systems acquisition. In addition to providing technical guidance in the areas of radar and EO/IR sensor simulation, environment simulation, and the development of digital data bases, Mr. Stengel served as the Lead Training Systems Engineer in the B-2 SPO. He received his BS in Aeronautics and Astronautics from New York University in 1971, an MA in Industrial Management from Central Michigan University in 1976, and an MS in Computer Science from the University of Dayton in 1984.

Thomas W. Hoog is the Technical Expert in the Training Systems Division of the Aeronautical Systems Division. Mr. Hoog has participated in the planning and completion of R&D projects as well as acquisition of operational systems, particularly in the areas of radar and EO/IR sensor simulation, EW simulation, environmental simulation, and data bases. He has held a variety of positions in the Training Systems SPO and ASD's functional engineering directorate. Prior to his present assignment he served as Chief Support Systems Engineer in the B-2 SPO. He received his BEE and ME from the University of Louisville in 1964 and 1975 respectively, and his MS from the University of Dayton in 1972.

ACTIVE SONAR CLASSIFICATION TRAINING USING RECORDED DATA

Dr. Leonard D. Healy
Naval Training Systems Center
Orlando, Florida

M. Gette Beauvais
Applied Research Laboratories
The University of Texas at Austin
Austin, Texas

ABSTRACT

The improved technology for quieting submarines has placed a renewed interest in the use of active sonar in Anti-Submarine Warfare (ASW). This increases the need for training ASW operators in the effective use of active sonar, particularly in the classification of an active sonar contact as either a target (submarine) or a non-target (e.g., a sea mount or a school of fish). Current trainers using synthetically generated contacts do not provide the realism needed for classification training. Operators have very little chance to practice using actual acoustic contacts.

This paper describes two demonstration models of a trainer using recorded acoustic contacts for active sonar classification training. The demonstration models were developed by Applied Research Laboratories, The University of Texas at Austin (ARL:UT), under the sponsorship of the Naval Training Systems Center (NTSC). The first is DEC MicroVAX based and is compatible with the hardware available in the Passive Acoustic Analysis Trainers (Devices 21H14 and 14E40). The second is personnel computer (PC) based and provides a much less expensive implementation. Both models provide the CRT display and the audio signal available in the operational sonar. The result is a trainer that provides the realism needed for classification training. The low cost of these units should make them applicable to the full spectrum of operator training, from classroom training to on-board refresher training.

INTRODUCTION

The ability to distinguish submarine targets from non-submarine targets is a critical skill required by the active-sonar operator. The operator determines whether a contact is valid by indications such as track consistency, echo shape, echo consistency, and particular aural characteristics. False contact indications include erratic track motion, no track motion, inconsistent echo quality, and non-submarine aural characteristics. Operator training requires accurate representation of the echo and aural characteristics to support target classification training and practice.

Use of Recorded Acoustic Data

The use of recorded data from actual acoustic contacts for active classification offers a means of providing a high fidelity presentation of aural and visual data needed for training. Using recorded acoustic contacts for active classification training is an extension of technique used in the Submarine Passive Acoustic Analysis Trainer (Device 21H14) and the Surface Passive Acoustic Analysis Trainer (Device 14E40)[1]. These trainers provide more than 100 student stations for hands-on classroom training in passive acoustic analysis.

Preprocessed recorded sonar data provides high fidelity displays and aural cues not previously available in sonar operator training. Conventional

trainers that electronically stimulate an actual sonar set do not produce the subtle signal characteristics needed for the classification task. Realistic aural signals are more difficult to synthesize than realistic visual displays. Passive sonar displays and aural signals generated by using stored data have proven effective in training passive acoustic analysis[2]. The same technique should be effective for teaching active sonar classification.

As a side benefit, the use of stored data produces a more affordable trainer. Preparing the actual sonar data for display in advance, using off-line signal processing, reduces the computation load on the trainer. This allows the use of a less expensive computer workstation or a personal computer. Low-cost desktop computers currently available provide the capability to replicate the operational displays, implement an instructional system, and store the processed data. The result is a trainer that offers the realism of actual sonar contacts in a portable trainer. The low cost and portability make possible wide distribution of trainers to fleet units and to reserve units.

The use of recorded data in active sonar is more complex than the application to passive sonar. In passive sonar, the entire signal processing necessary to prepare data for display can be performed off-line. In active sonar, a major part of the signal processing can be performed off-line, but some of the functions must be performed during the training exercise. The operator has choices such as range setting and filter bandwidth selection that make prestorage of every case impractical.

Capabilities Required

The following capabilities are required for an active sonar classification trainer:

- Fidelity. The acoustic signatures displayed must provide sufficient fidelity to support target classification proficiency training and practice.

- Synchronized Aural Signals. The aural signal presented to the student must provide equivalent realism to that required of the graphics display. In addition, the aural signal and graphics display must be synchronized so that the timing of the aural signal matches the display.

- Ease of Operation. The system must be user friendly so that the sonar operators do not need special training to use the trainer.

- Portability. The system must be easily transported on and off a ship. It must be small enough to be used in compact shipboard spaces.

- Automatic Evaluation. The system must automatically evaluate operator skill levels and provide guidance concerning additional training or practice that is required.

- Instructional Capability. The system must provide appropriate additional training based upon the particular operator's skill level. The process must be automated so that instructor intervention or participation is not required.

SONAR SYSTEM DESCRIPTION

The AN/SQS-53A[3] sonar set was selected for the Active Acoustic Analysis Demonstration Unit (AAADU). The AN/SQS-53A is the Navy's primary surface ship sonar and will remain so through the 1990s. Further, recorded sonar data are available for this unit.

Figure 1 shows a functional diagram of the AN/SQS-53A. The single time-shared transducer is used for both transmitting and receiving acoustic signals. The transducer transmits both continuous wave (cw) and coded pulse (CP) signals. There are separate beamformers for the variable depth receiver (VDR) and the surface duct receiver (SDR). The VDR processes both signal types, and the SDR processes only cw signals.

Displays

The receivers provide input to three consoles which have active sonar displays. The VDR detection (A-scan) console presents a display of range versus amplitude for 12 CP and 12 cw beams from the VDR. This display is used primarily for target detection with the VDR receiver. The SDR detection (B-scan) console presents data received on the 72 SDR beams on

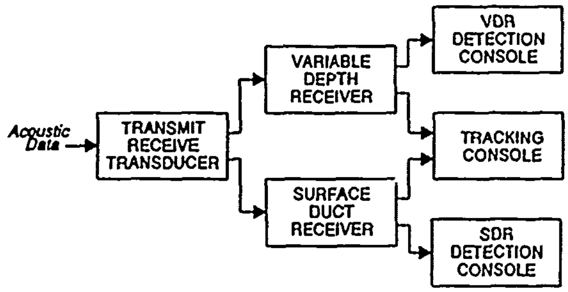


Figure 1. AN/SQS-53A Sonar Functional Diagram.

the plan position indicator (VDR) display and a B-scan display. The VDR display shows a polar presentation with the center representing own ship. The B-scan display shows range versus true or relative bearing.

The target tracking console can be used with an SDR or a VDR target. The console presents two displays: (1) the sector scan indicator (SSI) and (2) the target Doppler indicator (TDI), along with an aural signal. Figures 2 and 3 show these displays for cw and CP signals.

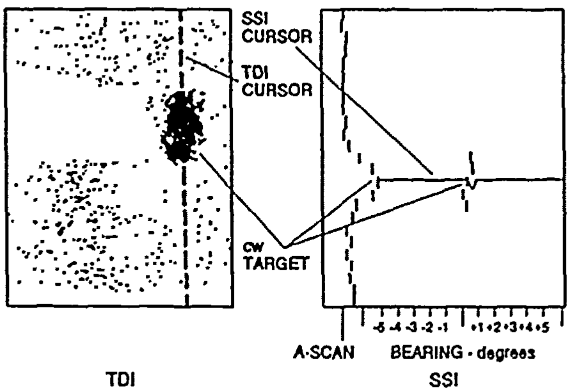


Figure 2. cw Target Tracking Console Display.

The SSI display provides two presentations. The A-scan on the left side shows processed signal amplitude in the horizontal direction versus range in the vertical direction. The SSI shows fine target bearing and target range values. The SSI cursor allows the operator to select a target by placing the range line with its movable notch over the return of interest.

The TDI display provides information from which an operator determines target Doppler and target Doppler consistency. The presentation shows target Doppler along the horizontal versus target range along the vertical axis. The range axis for the TDI and SSI displays is the same.

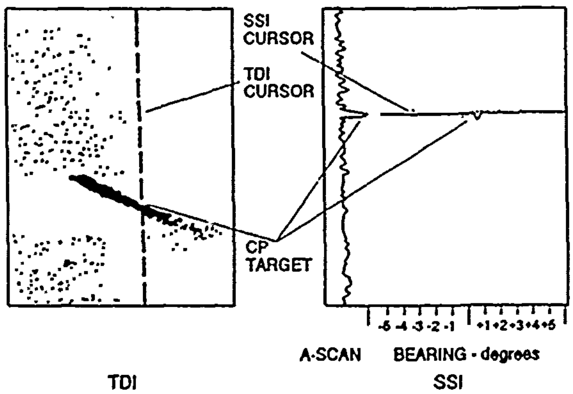


Figure 3. CP Target Tracking Console Display.

Display Selected for Training

The target tracking console displays were selected for implementation for the AAADU for the following reasons.

- The fine target bearing and range information provides the best resolution available for classifying active targets.
- The SSI display can be used to classify shortrange submarine targets by "linelikeness." That is, the echoes from individual scatterers along the submarine map out a line on the range-bearing display. This mode of classification is not available on the VDR or SDR display consoles, because only very low resolution target bearing information is available on those displays.
- The TDI display provides target speed information that is useful in classifying targets with significant opening or closing Doppler.

Operator Controls

The following controls are available to the operator at the tracking console.

Range Window. The range scale on the SSI display is operator selectable in three range increments: 2000, 1000, and 500 yards.

Pulse Mode. The transducer is capable of transmitting cw signals of varying lengths from 30 milliseconds to 1 second for SDR processing. For the VDR the sonar can transmit both a cw signal and a CP signal. The duration of the cw signal is variable from 10 milliseconds to 0.5 seconds. The CP signal is fixed at 0.5 seconds. The operator may also select various transmit sequences.

The transmit mode for the VDR is selected by the operator at the VDR detection console (or A-scan console). The transmit mode for the SDR is selected by the operator at the SDR console (or B-scan console). The tracking console operator options are limited to the selection of one of the available signals.

Doppler Filtering. The operator may select two optional narrow bandwidth filters (equivalent to ±2 knots and ±6 knots) to be applied to the TDI data before it is displayed.

Threshold. Nine threshold values are available to filter what data will be shown on the SSI display. The threshold values are operator selectable, but not from the front panel.

SIGNAL PROCESSING

The signal processing emulates the processing in the AN/SQS-53A sonar set[4]. Figure 4 shows a simplified block diagram of the target tracking receiver. The data from recorded contacts are stored after the beamformer. The signal processing functions are (1) signal conditioning, (2) target tracking signal processing, (3) display generation, and (4) audio generation. The data for each acoustic contact was recorded using whatever pulse mode was chosen by the operator during the exercises. The data selected for implementing the AAADU was limited to single pulse sequences consisting of both CP and cw signals.

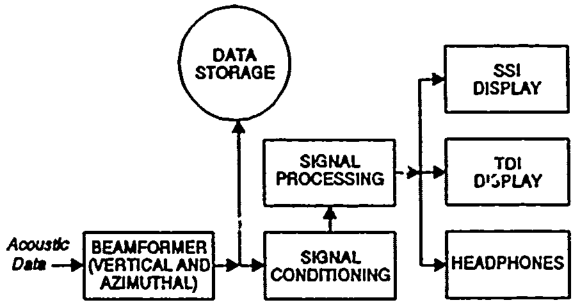


Figure 4. Target Tracking Receiver.

Signal Conditioning

Figure 5 shows the processes performed in signal conditioning for the AAADU. The processing converts the recorded data to complex data samples used by the VDR and SDR.

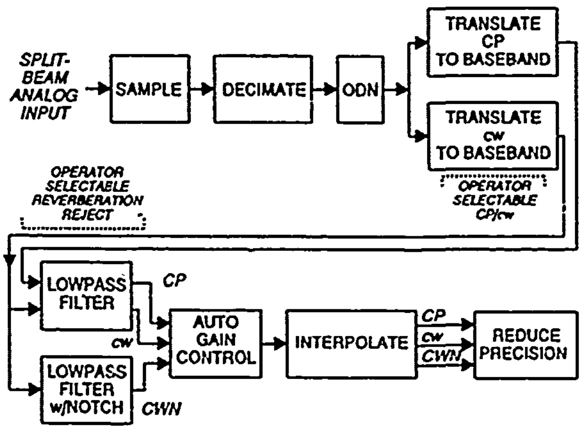


Figure 5. Signal Conditioning Implementation.

Sampling and Decimation. The first step is to convert the analog data to digital form. The data are sampled at four times the center frequency of the recorded signal, then the number of samples is reduced by taking only two adjacent samples out of every 24.

Correction for Own Ship Doppler. The Own (ship) Doppler Nullification (ODN) correction factor is obtained for the ship's trajectory and servo information recorded on the analog instrumentation tapes. The frequency shift due to the sensed own ship motion is computed, and

frequency shift due to this motion is eliminated by translating each complex data sample by the appropriate value.

CP/cw Separation and Notch Filtering. The CP and cw bands are converted to baseband frequency and separated by two low-pass filters into two data streams. A third data stream is produced by applying a notch filter equivalent to ± 6 knots to the cw data.

Automatic Gain Control (AGC). The AGC function is implemented by computing a gain value that makes the average input signal equal to a constant. The number of samples considered in setting the gain corresponds to the time constant selected for the pulse length being transmitted. The sampling interval is 0.15 seconds for pulses of 0.1 second or less, and 0.5 second for longer pulses.

Interpolation. The data for each data stream are interpolated to give the same output sampling rate as the complex sampler in the AN/SQS-53A. This introduces an error of less than 1 dB. in amplitude and less than 0.2 degrees in bearing. These errors are negligible compared to the resolution of the displays. After interpolation, the results are reduced to 8-bit samples to match the precision of the complex sampler in the AN/SQS-53A.

SSI Display Signal Processing

Figure 6 shows the signal processing to convert the conditioned CP, cw, and notched cw signals into an SSI display. The dotted line separates those processes that can be performed off-line from those that must be performed in the trainer.

Replica Correlator. Replica correlation is the matching of the received data to stored replicas of the transmitted signal. For CP signals, the correlation uses a single replica of either the up swept or down-swept FM transmitted signal. For the cw band, the reference signal is read out at various rates to produce shifted replicas to produce the effect of Doppler shift. The comparison of the replica with the signal is performed by a set of matched filters.

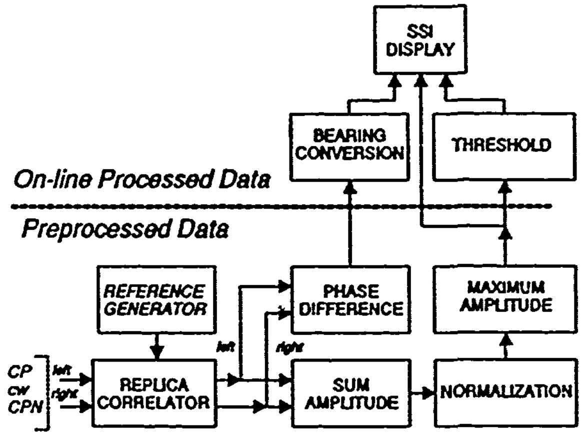


Figure 6. SSI Simulation.

Amplitude, Bearing, and Normalization. Left and right beam correlator output data are summed to form sum beam data. These data are normalized and the maximum amplitude over the matched filter outputs is found and reduced to a 6-bit representation. The left and right matched filter outputs, corresponding to the maximum amplitude, are used to compute a phase difference. This phase difference is reduced to a 10-bit bearing angle. This 16-bit representation of an amplitude and bearing for each range increment forms the data base for the AAADU.

Display Generation. The A-scan part of the SSI display presents amplitude of the return signal on the horizontal axis versus range on the vertical axis. The SSI display shows maximum signal return at each range at a vertical position corresponding to range and a horizontal position corresponding to bearing. The amplitude value determines the brightness of the point displayed. Those points whose amplitudes are less than the threshold set by the operator are not displayed.

TDI Signal Processing

Figure 7 shows the signal processing necessary to convert the conditioned CP, cw, and notched cw signals into a TDI display. The dotted line separates those processes that can be performed off line from those that must be performed in the trainer.

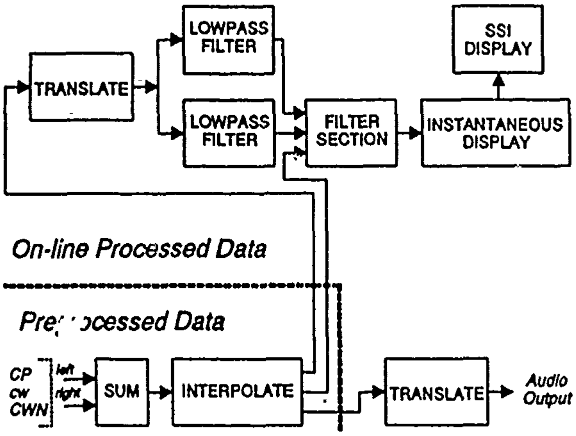


Figure 7. TDI Simulation.

Sum and Interpolate. Left and right split-beam data from the signal conditioner are summed and interpolated to produce a 800 Hz. sampling rate. The result is provided to the AAADU for use in generating the aural signal and the TDI display.

Filtering. Data for the TDI display are filtered to eliminate out-of-band signals according to the mode selected by the operator. One of three filters is applied to reduce the Doppler range of the target that may be displayed.

Instantaneous Frequency. The instantaneous frequency is computed from the rate of phase change over time. The frequency variation is then converted to target speed variation in knots.

Audio Signal Generation.

During display generation, the aural output is formed by translating the stored audio baseband data to a center frequency of 800 Hz. and converting the data to an analog signal. The 800 Hz. quadrature samples are converted to a 3,200 Hz. sampling rate and used to drive the headphones.

HARDWARE IMPLEMENTATION

Three sets of hardware are used in the demonstration of the active sonar classification trainer. The off-line signal processing hardware consists of analog-to-digital conversion equipment and a general-purpose computer. This hardware is used to process the AN/SQS-53A analog instrumentation tapes obtained from the Naval Underwater Systems Center, New London Laboratory (NUSC/NL).

AAADU Implementation.

The hardware selected for the AAADU is a MicroVAX II workstation with a high-resolution (1024 x 1280) graphic system. The MicroVAX II was chosen to make the AAADU hardware compatible with the 14E40 and 21H14 series of passive acoustic analysis trainers developed for the surface and subsurface communities.

Special equipment was added to implement the aural portion of the trainer. In the passive trainers, only one audio channel is required. A stereo audio cassette tape provides the aural signal on one channel and a timing track for synchronization on the other channel. The active sonar trainer requires two aural tracks, one for cw signals and one for CP signals.

The aural signal generation was solved by storing the data in digital form and converting the data to analog form to drive a speaker. After conversion to analog form, the aural data is filtered and played through the DECtalk speaker. The filtering is performed by the Q-bus interface board, designed by ARL:UT. The DECtalk voice printer was modified to allow the audio signal from the filter to be played through the DECtalk speaker. This configuration allows the audio signal to be mixed with text output to the voice printer. Headphones connected to the voice printer provide the aural signal for active classification.

Personal Computer (PC) Implementation

A PC demonstration unit was developed to take advantage of the PC's low cost, small size, light weight. The PC implementation provides essentially the same graphics display as the AAADU, even though the resolution is approximately one-half that of the high-resolution monitor used with the workstation. The PC version uses audio cassette tape to provide the aural signals.

The aural signals generated by the AAADU are recorded on one of the stereo channels of an audio cassette tape, and a timing track is recorded

on the other track. The timing track is used by the software program to synchronize the display with the recorded audio signal. Because only one channel is available for audio, it is not possible to change the audio as the operator switches between cw and CP signals.

DATABASE

The data currently available for use in the AAADU consist of analog recordings of sonar contacts from the AN/SQS-53A AN/SQS-26 VDR. The data sets are shown in Table I. The data consist of dedicated submarine operations, false targets of opportunity, biologics, wrecks, bathymetric features, and surface contacts.

Table I. Data Sources.

DATA SET	ESTIMATED TOTAL PINGS	DATA TYPE
SPRUANCE	555	SUBMARINES FALSE TARGETS
McCANDLESS	1273	FALSE TARGETS
CUSHING/O'BRIEN	850	DD963 DESTROYERS
CUSHING/HOUSTON	2300	DEEP WATER 688 CLASS SUBMARINES
COMPTUEX 1-87	474	SHALLOW WATER SUBMARINES SURFACE SHIPS
GLOVER ASW 1987	2890	CONVERGENCE ZONE SURFACE DUCT SUBMARINES NON-SUBMARINES
I-SHAREM 1-87	2800	DEEP WATER SUBMARINES
ASWEX 86-2	400+	SHALLOW WATER SURFACE SHIPS
ASWEX 86-4	400+	SURFACE SHIPS FALSE TARGETS

Data Requirements

Training requires two sets of data: a teaching set and a testing set. The teaching set must show clear examples of both true and false targets in a low-level noise background. The false targets should include surface ships, wakes, kelp beds, and biologics. Examples of false alarms from reverberation, ambient or self noise, and slamming or quenching of the sonar dome should be included in the training set. The trainee should be taught to distinguish between submarine targets and commonly occurring returns that represent false alarms.

The teaching set and the test set should each contain approximately 100 examples of submarine targets and 100 examples of false targets. Each example should provide 15-30 pings on a particular contact. Assuming an average ping length of 0.5 seconds, 50-100 hours of active data are required for training and evaluation.

Noise Background

The ambient noise and/or background level has a significant effect upon the difficulty of classification. Testing of a trainee's skills before and after training requires a data set which various signal-to-noise (S/N) ratios. The different S/N ratios are necessary to provide graded test conditions to bracket the trainee's abilities. The test set must provide a high S/N to provide success, decreasing to a low S/N to provide challenge. It should be noted that a S/N ratio low enough to cause marginal trainee performance may be below the detection capability with either the SDR or the VDR.

Sea data with controlled S/N is difficult, if not impossible, to obtain. However, suitable data can be created artificially without destroying the realism of the data. This is accomplished by summing target returns with ambient/self noise or reverberation. The S/N is controlled by changing the level of the interference before combining the two sets of data. The summation of data requires the following conditions.

- Each input must be sampled at the same rate and decimated by the same factor.

- Each set of data must be translated to baseband (if not already there) and individually corrected for own ship Doppler.

- The data must be summed before the AGC.

It is advisable, but not required, to combine data before the low-pass filter to assure that the same filter is applied to each set of data. Note, however, that the power level of the individual sets of data must be determined after the low-pass filter. Also, the sonar transmit and receive configurations must be the same for each of the data sets combined. For example, cw echoes should not be combined with CP reverberations.

SUMMARY

The primary objectives of this project were (1) to develop a more effective trainer for active sonar classification by using preprocessed recorded sonar data to produce realistic displays and aural cues, and (2) to develop a more affordable trainer by using relatively inexpensive microcomputer technology.

The displays from the target tracking console of the AN/SQS-53A were selected as the most appropriate displays for training sonar operators in the task of classification. This selection is based upon the higher resolution presentation resulting from the fine target bearing and range information obtained from the split-beam processing.

Implementation of the AAADU included (1) preprocessing and storing data recorded from the VDR of the AN/SQS-53A sonar, (2) development of hardware and software to play back the preprocessed data in realtime on simulated displays of the target tracking console. The presentation includes visual displays like those of the tracking console and audio played through headphones or a speaker. Operator input is provided by a trackball.

Approximately 2,000 pings of recorded data were preprocessed. The scenarios available include submarine targets as well as false targets such as sea mounts, wrecks, and whales. The data base available is sufficient to eva'uate the training capability of this device. A larger database is needed for actual training.

This development has demonstrated that the use of preprocessed data and a relatively inexpensive microcomputer can provide an effective and affordable active classification training capability. The realistic displays and aural cues provided by use of prestored data can improve an operator's ability to discriminate between submarine and non-submarine signal returns.

The low cost of these microcomputer based trainers should make them applicable to the full spectrum of operator training. High fidelity active classification training can be added to the present classroom training. Refresher training can be moved from the schoolhouse to shipboard. Continuous refresher training should yield a significant improvement operator performance.

REFERENCES

1. Healy, Leonard D., *et al.*, "A New Generation of Trainers for Acoustic Analysis," *Proceedings of the Eighth Annual Industry/Interservice Training Equipment Conference*, November, 1986.

2. Pfeiffer, Mark G., and William C. Rankin, *Training Effectiveness Evaluation of Passive Acoustic Analysis Trainer (Device 21H14)*. (NTSC Technical Report 86-019) Orlando, FL: Naval Training Systems Center, October 1986.

3. NUSC Technical Document 5351-II, *Operational Doctrine for the AN/SQS-53 Sonar System, Vol II.* (Declassified 15 Mar 1988).

4. Beauvais, M. Gette, Maurice Mahlab, and J. F. England, *Signal Processing and Display Technology and the Active Acoustic Analysis Demonstration Unit*. (ARL Technical Report 84-46) Austin, TX: Applied Research Laboratories, The University of Texas, August 1987.

ABOUT THE AUTHORS

DR. LEONARD D. HEALY is an Electronics Engineer and Acquisition Director with the Naval Training Systems Center. He is

currently responsible for the Center's research program in Anti-Submarine Warfare Training. He holds degrees in Electrical Engineering from the University of Florida (PhD), Stanford University (Engineer), the Massachusetts Institute of Technology (SM), and the Georgia Institute of Technology (BEE), and an MS in Management from Rollins College. In earlier associations he was a Design Engineer and Section Head at Martin-Marietta and a Senior Research Scientist at Lockheed Missile and Space Company, where he was responsible for the design and development of special-purpose digital computer systems for air defense systems and missile checkout.

M. GETTE BEAUVAIS is a Research Scientist Associate with Applied Research Laboratories, The University of Texas at Austin (ARL:UT). She is responsible for several programs relating to automated detection, classification, and localization of active sonar contacts for both tactical and surveillance systems, and also relating to operator active sonar analysis training. She received a B.A. in Mathematics from the University of Texas in 1973 and has been employed at ARL:UT since 1976. Her previous work at ARL:UT has been in the areas of Torpedo Guidance and Control, Torpedo Defense, and Dynamic Engineering Modeling.

A LOW-COST/HIGH PERFORMANCE SENSOR SIMULATION: THE NEXT GENERATION

John Burkley
Andrew Gurcak
Loral Defense Systems-Akron
Akron, Ohio

ABSTRACT

This paper describes a real time Sensor Simulation system that utilizes an array of processors organized in a fine-grain multi-instruction multi-data stream (MIMD) computer architecture. The application described here is for a multi-mode radar simulation. The software was developed using the structural model with coding in Ada. The methodology for implementing a radar simulation on this architecture, database storage, and software development approaches will be discussed.

INTRODUCTION

Simulation of new sensors, including synthetic aperture radar and imaging sensors, demands high fidelity imagery at increased resolution. Attempting to meet the requirements of these systems simply by brute force increase of processing power and data storage leads to unwieldy increases in database storage and real-time computing requirements. Future training systems will be both deficient and more expensive unless affordable alternative solutions are found. These new systems must also be highly reliable and offer reduced lifecycle costs. To meet these challenges the following must be addressed:

1. New computer architectures offering increased performance at lower cost.

2. New sensor simulation architectures that take advantage of the latest processor and memory technology to meet computation and on-line storage requirements in a cost-effective manner.

3. New database concepts and effective data compression/expansion techniques.

4. Innovative algorithms that allow processing of all sensors on a common platform with a common database.

5. Software implementation using object-oriented design techniques and coded in Ada.

The Loral NOdal Sensor Simulator (LNOSS) described in this paper meets these challenges. It consists of "n" RISC processors (Intel 80960MC) organized in a linear array and programmed using Ada. The radar simulation problem was chosen as a test case to prove this design's applicability to generic sensor problems. The focus of this paper is a description of this design and a solution to the radar simulation problem. The algorithm supports data compression while maintaining full database information and fidelity. The software was designed using the structural model with all coded in Ada. We believe the techniques described in this example are also applicable to higher rate EO/IR sensors and 3D laser sensors.

ARCHITECTURE

We performed an architecture study to recommend advanced computer architectures for sensor processing. This study found that the radar simulation problem has inherent parallelism and that only minimal computational node interaction was required. It found that the work function per database element varied significantly. Finally, it found that a global communication between nodes would be required. These results suggested that a fine-grain multi-instruction, multi-

data stream (MIMD) computer architecture organized as a linear array would be a good fit for this class of problem. Each node in the array consists of a RISC processor with sufficient processing and memory capacity to support execution of the full sensor problem on a limited set of data in real time. For inter-node communication we chose a Time Multiplexed bus using a token passing approach. This minimizes hardware and has sufficient bandwidth for medium size networks. A VME bus I/F is also provided for global communication with the system controller and database update from disk.

DATABASE CONSIDERATIONS

The foundation of the imaging system is the gridded database whose spacings are tuned to the resolution requirements of the display device itself. The spatial frequency of the grid sampling is dictated by the "information content" (i.e., pixel) on the screen. The minimum information content is that required to just distinguish or, in visual terms, resolve two closely spaced high contrast point radar targets on a pixelized display. The thrust then becomes arriving at that minimal set of informational parameters for an area of terrain and features which, when formed into a grid unit (datapost), can be processed by the real-time software into the range-azimuth display area covered by a single pixel. Each radar display represents a collation of pixels for a particular range, thus implying the formation of a set of databases, each designed for a particular range. The on-line digital database is a collection of radar information that is stored in logical planes corresponding to each range (resolution). For the entire gaming area, each plane is a set of disk storage blocks, modified appropriately for the sampling integerizations induced by latitudinal variations in geometry. A critical feature of this form of on-line database organization is that processing is automatically normalized for all imaging work, i.e., the real-time system is always processing the same number of dataposts for any resolution, thus reducing the processing capacity required.

DATAPOST CODING

The radar-significant elements of each individual sampling square are encoded into the digital radar landmass system (DRLMS) datapost. The function of the datapost is to convey to the radar processing system the amount of radar energy that a terrain (including features) square reflects back to the aircraft's antenna. A single average reflectivity assigned to a terrain square is not sufficient to do this because the actual reflectivity is dependent on many terrain and aircraft sensor related factors. The primary topographical factors governing radar behavior are the physical composition of the terrain square and its orientation with respect to the radar source. Physical composition includes the general topographical makeup and the makeup of any structures that may be located within that square. For any (instantaneous) fixed position of the source, the terrain height and slant range, with respect to the source, determine the relative radar illumination angle. If there are structures on the terrain square, their height, shape, distribution over the square, and orientation all determine the relative illumination angle. Structures frequently have flat surfaces that exhibit very strong reflectivity at nearly perpendicular illumination angles, and little at other angles. For this reason, the model operates with base reflectivities over the broad range of grazing angles, with special augmentation for those reflectors that exhibit pronounced specular or low-grazing-angle reversal effects. Each datapost is divided into descriptive fields, including terrain and feature height, reflectivity, feature character and directivity, and special orientation, shadow, and coverage codes.

ALGORITHM DESCRIPTION

The basic algorithm used to implement a radar simulation is based on a pipelined station concept developed for the F-15E DRLMS in which the problem is broken into nine steps or stations (see Figure 1). The following paragraphs describe each stage of the software pipeline:

1) System Controller: Although not part of the radar sequence proper, this task determines whether the node owns a particular radial, and monitors the loading of the database memory needed for the picture.

2) Radar-Polar: This task forms the string of dataposts from the nadir to range from the appropriate resolution plane into a "radial," which becomes the basic processing unit for the remainder of the algorithms.

3,4) Radterp/Crossterp. This process unpacks

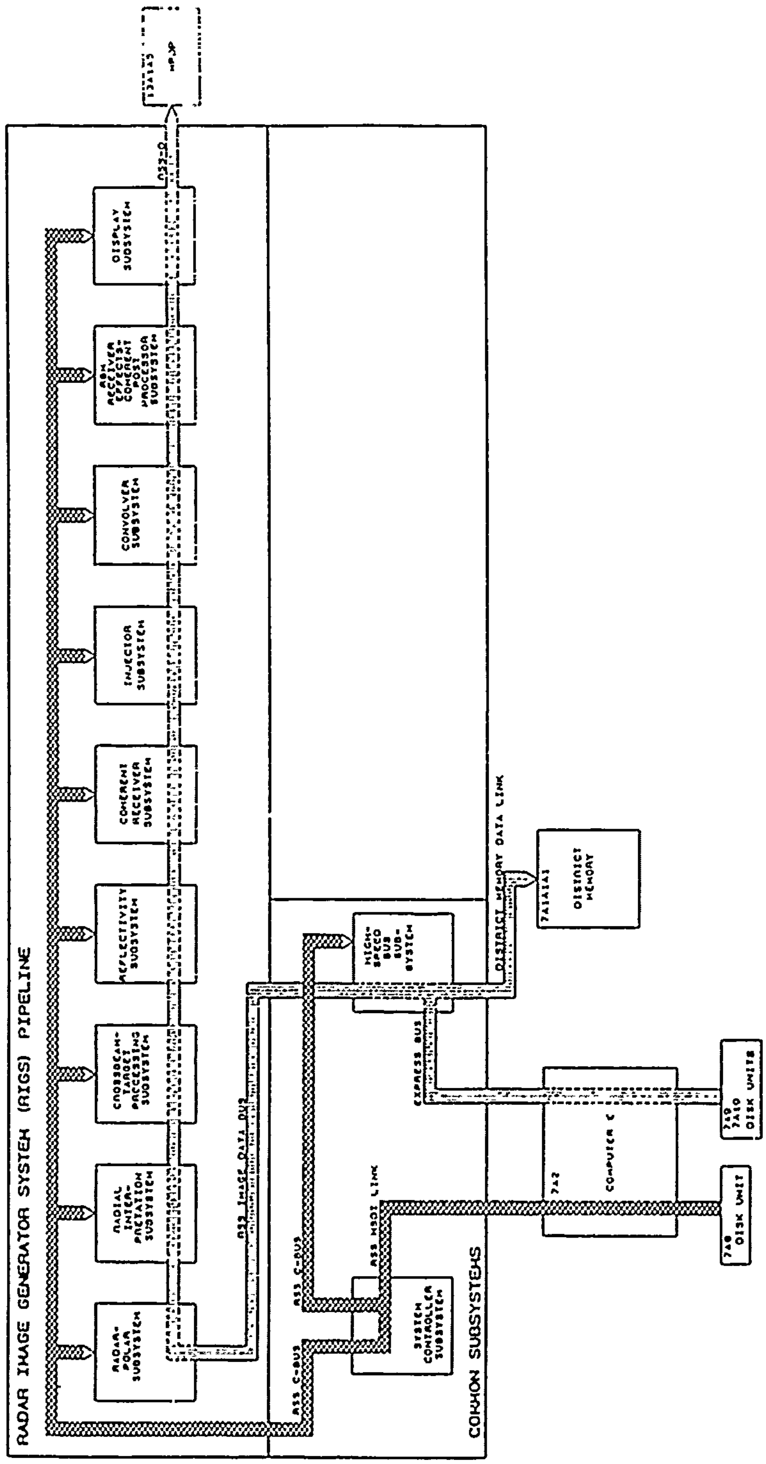


Figure 1 — F-15E DRLMS Pipeline Basic Block Diagram

the datapost words into fields for further use. It expands the feature height data and generates various along-range effects, as well as signalling the possibility of low-grazing-angle effects processing. It also buffers several radials to discover cross-range effects. It processes cross-range terrain-tilt, presence and orientation of cardinal reflectivity effects, and the interpolation of cross beam terrain height.

5) Reflectivity: This process induces a multitude of radar effects into the image. It determines the radar backscatter of landmass elements for terrain, and feature tops and sides. By use of the separability of return calculations, it computes layover, moving target displacements, and height displacements as required. In terms of aspect effects, it performs horizontal and vertical aspect calculations for terrain and features, taking into consideration such non-Lambertian variations as low-level/grazing angle effects, plateau/ medium angle effects, and specular/high grazing angle effects so that basic slope, onset, leading edge, and low-level effects are directly calculated. In addition to setting driving mechanisms for INJECTOR, it also handles imaging geometry attenuations with respect to range and vertical antenna patterns.

6) Injector: The injector implements efficiency in spatial domain processing, the system impulse response function for each significant elementary return. This process begins by calculating the centroid displacements of features resulting from datapost expansion. From the expanded data-posts, such map distortions as range, range rate, and cumulative system mapping errors are determined and placement adjustments made. The spread function yields the basic aberration for range and azimuth focus, the range and azimuth displacement from sidelobe levels, and the mainlobe loss and sidelobe increase from energy conservation and amplitude management. Trace aberrations involving range and azimuth walk and layover are also calculated. Extended special forms for range and azimuth ambiguities and systematic phase modulations are also performed. Speckle, clutter, and return fluctuations for PRF simulation are developed by partially random spread addressing formulations.

7,8) Convolver/Receiver. Beamwidth convolution and effects due to receiver noise, jamming and ECM/ECCM and feedback sampling are performed here.

9) Display Processing: Display formatting and processing are performed both on individual radials and on the map ensemble.

The new approach was not to redesign this basic algorithm, but rather to determine how it could be applied to an array of processing nodes. In a pipeline processor all the data is passed between the processors, and each performs a part of the problem. In parallel processing each processor solves the whole problem for a portion of the data. The number of processors required depends on the amount of data needed to be processed rather than the number of steps in the problem. The task at hand is to devise a method for distributing the data so that all the processors are kept busy. We solved this problem by applying the following techniques:

a. Radial Processing Distribution

We determined that the best way to share the work function was to operate on a number of radials at once so that each processor would have the same number of radials on which to operate independent of aircraft position. This was done by eliminating the database "District Memory" as a separate module and distributing the database storage among the processors.

b. Common Node Software

Each node contains the same software for the whole radar problem. Each node runs the same software on its set of dataposts and a broadcast set of initial conditions. Each node processes asynchronously approximating a MIMD processor; this tends to equalize the loading since the work function required to process a given post may vary.

c. Tier I/O

The radar problem has inherent parallelism but also has sequential processing dependent on intermediate values. In the pipeline technique, this intermediate data was added to the data post as it was passed between stages in the pipeline. We solved this problem in parallel by establishing tiers of processing after which each processor would

exchange datapost with all the other processors. We determined that three such tiers were required to solve this problem. The processor interface design was optimized to allow this data interchange at a high rate so that the total I/O time was less that 20% of available real time.

d. Display Processing

In the F-15E radar simulator, all display functions were handled by a custom display processor which took the output of the pipeline and performed coordinate conversion display formatting and storage in frame buffers for display. This operation can be handled much more efficiently in parallel. The video memory is partitioned across the processors. Also, the required video sync is imbedded in the video memory, allowing a generic video design to interface with a number of video standards simply by changing software. The display formatting and coordinate conversion is done within the processing node. Video memory is dual ported and scanned as needed by the video display controller.

LNOSS HARDWARE DESCRIPTION

The LNOSS system is a MIMD processor array of RISC processors. A detailed block diagram is shown in Figure 2. The heart of the system is a processor board which integrates an array of four RISC processors, supports a 64Mbyte/second interface bus, and provides VME bus compatibility as shown in Figure 3. The system assumes a VME form factor and has an open architecture which supports commercial off-the-shelf equipment.

DRLMS System Controller - A VME 68030 processor card will perform the GEO/Disk control, target processing, and system controller (SYSCON) functions. This card serves as VME bus master and has a serial RS232 interface for diagnostics and stand-alone operation.

LNOSS Processor - The LNOSS processor board performs the radar imaging tasks in the radar simulation pipeline. It consists of an array of four loosely coupled RISC-based 32-bit processors. The Intel 80960MC processor family was selected as the best of twelve candidates for serving as a node processor in this design because it was a single chip processor and required a minimum number of chips for a node, it had a verified Ada compiler as well as an impressive set of software tools available, and it is one of the JIAWG-approved 32-bit architectures. The LNOSS is built to a VME 6U 220mm form-factor with connectors P1 and P2 compliant to the IEEE P1014 VME standard. Each node provides a standard 32-bit VME port and an AP-port for highspeed (64Mbytes per second) interprocessor communication on available P2 connector pins. Each node includes 512Kbytes of SRAM for temporary storage and zero-wait state program storage, 1Mbyte of Flash EEPROM for processor boot, test programs, and other noncritical program and table storage, and 4Mbytes of DRAM for database, video buffer, and other data and programs as required. The design supports zero-wait state operation from SRAM. The DRAM memory is dual ported to both the nodal processor and the VME, allowing update of global parameters and database in parallel with nodal processing. A special 8-bit microcontroller is included on each PCB for I/O control and test functions. The number of LNOSS processor cards needed varies with the simulation requirements.

Disk Subsystem - The GEO/Disk subsystem provides geographical data to the radar pipelines from the on-line GEO data disk. The VME 68030 processor card prefetches database updates as required by the simulation problem. The GEO/Disk subsystem can support up to eight removable winchester disk drives on a SCSI bus. An 8mm tape drive is provided for database loading and system backup.

Display Interface - The display interface board is a simple frame buffer with a built-in scanner to perform the digital scan conversion. The display board itself has no processing capability; all display functions are done in the processor. The design is capable of supporting multiple images at once. One innovation is the use of imbedded sync in the frame buffer which the scanner reads out as data. This allows the output to be easily changed from one display format to another without changing hardware.

Reliability/Fault Tolerance - Because of its small size and high level of integration, the LNOSS processor has a predicted mission critical MTBF

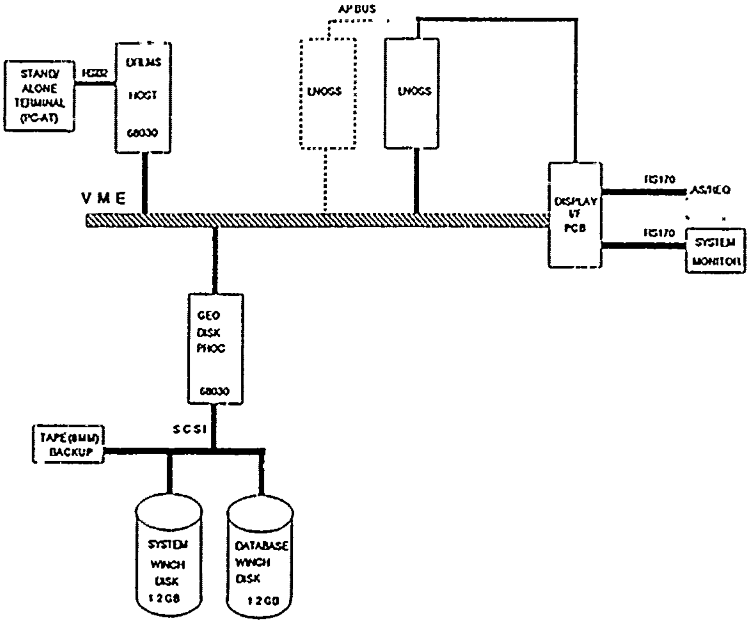


Figure 2 — LNOSS System Block Diagram

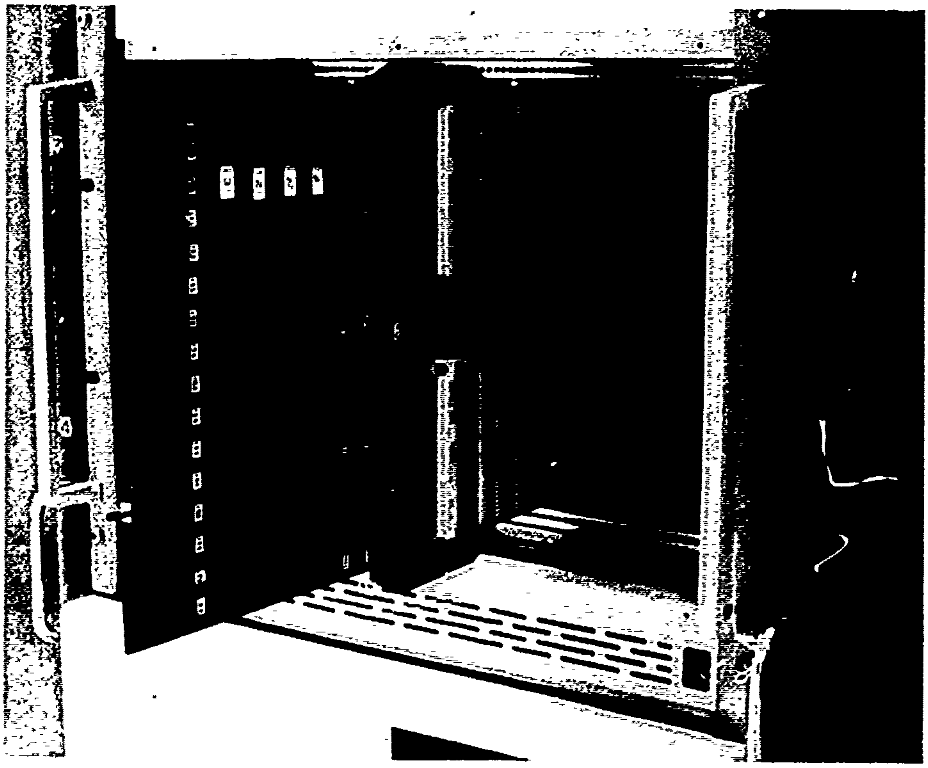


Figure 3 — LNOSS Processor Board

of over 4000 hours for a four-board system. The nature of radar simulation is that it is inherently fault tolerant to data errors. Because of the large amount of data and inherent smoothing operations in solving the radar equation, single bit errors from the disks and data memory do not constitute mission failures. The LNOSS design takes advantage of this characteristic of radar processing in its parallel architecture. The processing of the radar picture is scattered across the processor so that the failure of a single node is analogous to a data fault and does not cause the picture to degrade noticeably. The design of the system uses an on-board 8-bit microcontroller for rapid on-line identification and disabling of failed processors, and thus provides a significant fault tolerant capability.

SOFTWARE DESCRIPTION

The software has been designed to meet the goals of high-precision, real-time radar simulation and the DOD-mandated use of Ada for all real-time applications. These objectives have been met by keeping the fast, image-making portion of the real-time system generic and parameter-driven; all the specific details of beam-shaping, antenna characteristics, and processing effects can be driven by relatively few parameters calculated at a "slow" rate in the system controller. The real-time image-making software runs at maximum speed and efficiency, using Ada code for the LNOSS's Intel 80690MC microprocessors.

The radar simulation is organized along the structural model developed by the Software Engineering Institute under Air Force contract specifically for simulation. A structure diagram is shown in Figure 4. Its basic strategy is to employ a series of generic and specific software objects, with coupling among them minimized. The aim is to produce a "flat" system with as shallow a calling-tree as possible. Objects, both internally and in their interfaces, are based on a common structure or template, preventing software designers from arriving at conflicting decisions on the nature of their programs. It also ensures a uniform ease of documentation both for initial production and for subsequent modifications. It is frame-oriented in the system controller, but runs asynchronously in the LNOSS processor.

The radar image is formed by a multi-step process. The basic strategy is to direct the LNOSS processors to perform a series of ray traces from the instantaneous antenna position to the intersecting point on the earth along an arc from the aircraft nadir to the range limit, with the arcs (radials) mosaicked to form the picture following the antenna's motion. The basic geometry of the map is laid down by the DRLMS host at map initiation, and is continually corrected during the process to ensure natural tracking of non-linear scan rates for shearing and Doppler shift effects. During the map initiation process, coarse target screening is performed to see if any moving targets may be present in, or move into, the current map. The function of the system controller is to supervise the I/O transfers between the trainer and itself, calculate "slow" parameters, and prepare information buffers for the image-making system of distributed microprocessors. The LNOSS scheduler is a locally data-driven format. As the DRLMS host data is broadcast to all LNOSS nodes, each of them selects the radials it "owns" and works on the imaging process, bringing it to completion as the displayed radial.

CONCLUSIONS

The results to date show the capability to provide a demonstratable radar image on a single LNOSS card. A system of multiple LNOSS boards will provide a fully compliant digital radar landmass system programmed in Ada. This system equals the performance of the F-15E DRLMS at an order of magnitude lower cost. We are developing a next-generation sensor processor which has the capability to provide high fidelity sensor simulation at an acceptable cost. We feel this design can be extended beyond radar sensors to EO and IR type sensors.

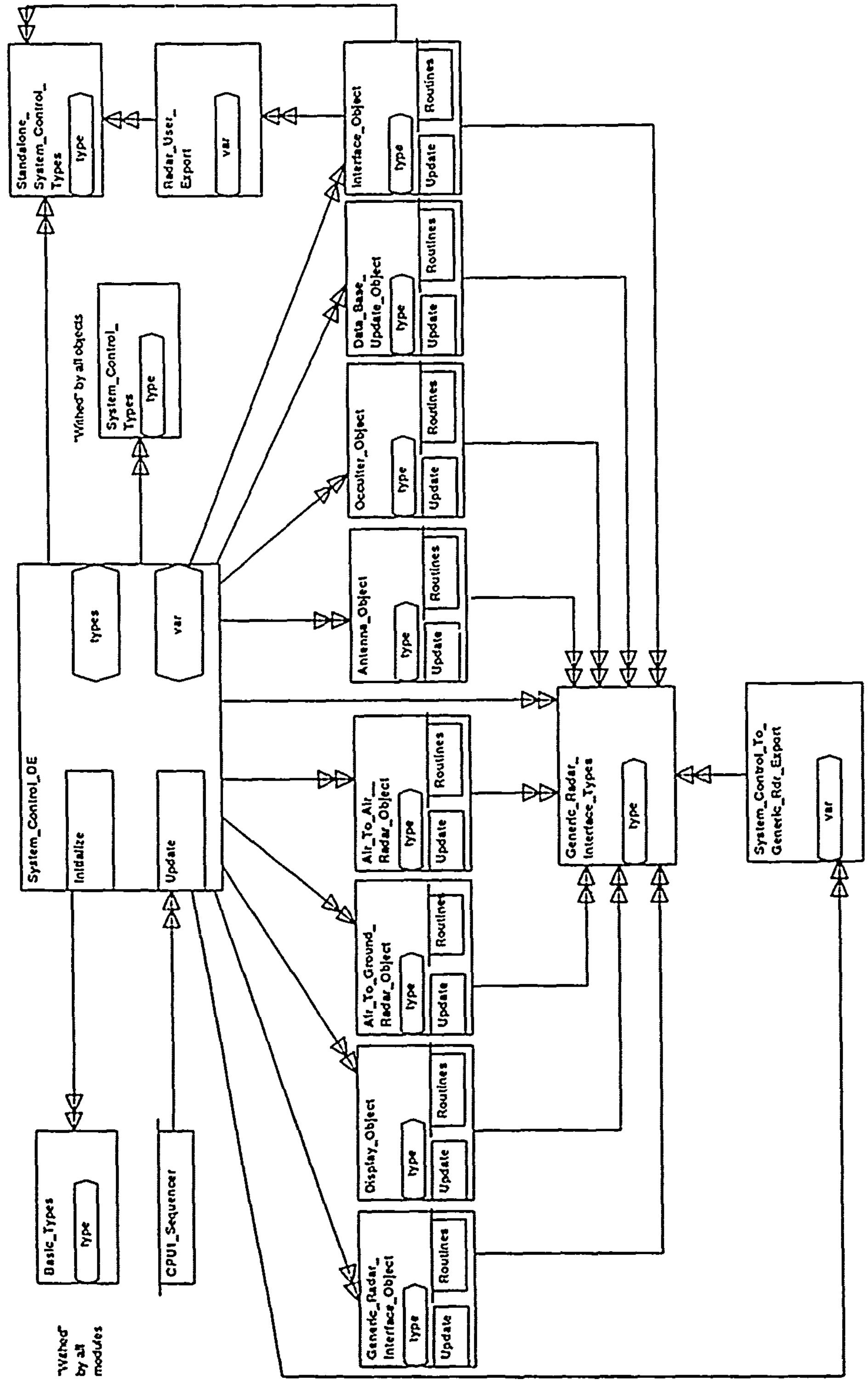


Figure 4 — DRLMS Host Real-Time Software
Top-Level Diagram

ACKNOWLEDGMENTS

The authors sincerely thank all those who have toiled lo, these many years in all incarnations of the Loral DRLMS. The authors especially wish to acknowledge the guidance of Lowell Stoner, without whom none of this would exist.

ABOUT THE AUTHORS

John Burkley is currently a member of the Senior Technical Staff, responsible for Sensor Simulation in the Simulation and Training Department at Loral Defense Systems - Akron. Prior to that, he was the Lead Hardware Engineer for sensor systems on the F-15E WST. He also worked on the Massively Parallel Processor delivered to NASA in 1983. Mr Burkley has a BSEE and MBA from the University of Akron and an MSEE from The University of Maryland.

Andrew Gurcak is the Responsible Engineer for IR&D in Training and Simulation at Loral Defense Systems - Akron. Prior to his current job, he was the Lead Software Engineer for sensor systems on the F-15E WST. Mr. Gurcak has a BS from MIT, an MA from the University of Virginia, and an MBA from the University of Pittsburgh.

GUIDELINES FOR EMBEDDED TRAINING DECISIONS

Bob G. Witmer & Bruce W. Knerr
U.S. Army Research Institute
PM TRADE Field Unit
Orlando, Florida

ABSTRACT

While Army policy requires training developers to consider embedded training (ET) first and foremost among training options, effective implementation of this policy has been hampered by the lack of specific procedures for determining what to embed early in prime system development. This paper describes specific procedures that assist a user in making those early ET decisions. Although task information has traditionally been the primary criterion used in selecting media for training, it is thought to be less important in deciding when to use ET than are the following factors: policy; system availability for training; the technical feasibility of ET implementation; the effects of ET on system reliability, availability, and maintainability; the impact of ET on system manpower and personnel requirements; the need for training-specific interface hardware; safety; and cost-effectiveness. These factors are incorporated in three sets of flowcharts, designed to be used in different stages of the acquisition process.

INTRODUCTION

Embedded Training (ET) is a training capability that is built into an operational system and requires access to and use of that system to conduct training. An Embedded Training System (ETS) is that part of the training system that includes the embedded training capability. While the concept of ET has been in existence for some time, instances of its successful implementation in Army systems are relatively rare. The emphasis on ET is increasing, however, as a result of several changes in Army policy, practice, and weapons systems. First, realistic unit training is being emphasized as a means to better prepare our forces for combat. Second, overall cost reduction has become mandatory while many of the costs associated with the operational use of the weapons system for training, such as increasingly powerful and sophisticated ammunition and the ranges on which it can safely be fired, are increasing. Third, more systems have embedded computer capability, which can support training if designed appropriately.

Aware of these factors, the Vice Chief of Staff, Army, and the Under Secretary of the Army stated as policy in March 1987, "An embedded training capability will be thoroughly evaluated and considered as the preferred alternative among other approaches to the incorporation of training sub-systems in the development and follow on Product Improvement Programs of all Army materiel systems."(2)

However, effective implementation of this policy has been hampered by the lack of specific procedures for making early decisions about what training to embed and what to provide by other means. This paper describes the development of a guide to help users to determine, early in the acquisition process, what training to embed into the prime system. Prime system refers to the operational system for which the training, embedded or otherwise, is required.

PROBLEM

Historically, training media selection decisions have been based on prime system design characteristics and the nature of the tasks to be trained. However ET poses unique problems for decision makers in that ET requirements must be determined early enough to be included in the prime system design. A Stand-Alone Device (SAD) can be based on prime system characteristics because the concept formulation process for the SAD training system typically lags the concept formulation for the prime system. ET, in contrast, is a part of the design of the prime system itself and its concept formulation and design must proceed concurrently. Furthermore, the task level information historically used to make training decisions is usually not available in time to be of much use in making ET decisions.

A BRIEF HISTORY OF PREVIOUS WORK

ET has been used in limited applications for at least three decades(6), but did not receive widespread attention until the 1980's. The 1980's were characterized by flurry of ET research activity. ARI, PM TRADE, and their contractors were highly productive, producing a ten-volume set of guidelines and procedures designed to support the effective consideration, definition, development and integration of ET capabilities. In addition to these documents, several ET surveys and literature reviews were completed, and development and evaluation studies were conducted for several systems including the Fiber Optic Guided Missile and the Howitzer Improvement Program.(4) While these guidelines provided detailed procedures for making ET decisions late in the acquisition process, they provided only general information for making decisions early in the acquisition process.

Researchers at the Naval Training Systems Center were also busy during this time, but their primary focus was on identifying design guidelines for effective ET for shipboard and other Navy

systems.[1,7] Meanwhile, work was also proceeding in identifying requirements for aircrew ET applications.[8] More recently, Eagle Technology, Inc. and Vector Research, Inc., under contract to ARI, initiated the development of an Embedded Training Candidates Model for determining, very early in the weapon system development process, the feasibility and value of including ET capabilities in the weapon system.[3] Although this work was technically sound, it was terminated while in a preliminary stage and was never formally published. A later effort by the same organizations[5] resulted in a design architecture for a decision support system for making early training strategy decisions, including ET. Again, while the work was technically sound, the next step, that of developing the functional specifications for the system, was never undertaken.

EMBEDDED TRAINING SYSTEM CHARACTERISTICS

To maximize training effectiveness, an Embedded Training System (ETS) must be a well integrated component of the total training system. All of the training needs for a given system, individual or unit are not likely to best be satisfied by ET. The ETS should therefore train only those tasks, functions and missions to which its characteristics are best suited. Other training media should be used where they can train more effectively than ET or train equally well at a lower cost. ETS effectiveness also depends on the incorporation of the following training features: a means of assessing student performance; a means of providing feedback to the student to reinforce and improve correct performance; and a means of record keeping, to allow the management of individual and collective training and identify deficiencies requiring additional training.[10]

The typical ETS is a computer-based system, either integral to or adjunct to the prime system, which, when activated, interrupts or overlays the system's normal operational mode to enter a training and assessment mode. The ETS also includes the facilities, expendable supplies and materials, and personnel required to provide embedded training. Although embedded training system designs can assume many different forms, they share the common characteristic that the student is trained using the actual controls and displays of the actual equipment. They differ along a continuum in the extent to which the ETS is fully contained within the prime system. These guidelines consider three types of ETS, defined below, that represent discrete points along an ET continuum that includes a potentially unlimited number of ET architectural types.

Fully Embedded

All training features, except for perhaps easily installed training software or courseware, are fully contained in the prime system itself. They go to war with the system. They meet the prime system Reliability, Availability, and Maintainability (RAM) requirements. A fully embedded ETS, on a vehicle, could train while the vehicle is moving, as in tactical engagement simulation. Fully embedded training is usually distributed with the prime system on a "one for one" basis.

Appended ("Strap-On")

Components of an appended ETS can be installed on or attached to the prime system when needed, and removed when they are not. An appended ETS will nevertheless require permanent, designed-in, components (such as sensors, mounting brackets, and connectors). An appended ETS could be used in assembly areas or in close proximity to combat. It could go to war with the system if it were so designed, although that is not a necessary characteristic of an appended ETS. It could train "on the move." Ruggedization may be required. One appended ETS could serve multiple prime systems, but could serve only one at any given time.

Umbilical

The umbilical ETS is similar to the appended system, but involves, in addition, physical connection(s) to external components, such as a computer, communications system, or Instructor/Operator console. As with an appended ETS, it requires some built-in features to interface with the external components of the system. An umbilical ETS may interconnect many systems, as in simulated networking for force-on-force training. The umbilical ETS is not a go-to-war training system. It cannot train "on the move." Ruggedization is unlikely to be required. One umbilical ETS can serve multiple prime systems.

Since these types differ along a continuum, it is possible to conceive of an ETS which is not easily classified, such as an ETS with an on-board ET component which communicates with an external component via radio or infrared transmission, rather than through a physical connection.

PROCEDURE

The guide for early ET decisions has been developed in accordance with the following principles. First, the decision process must be phased and linked to information availability. Tentative decisions must be made initially, and then revised as more information becomes available. Second, early decisions should be biased in favor of the use of ET, because it is easier to delete a requirement for ET than to add one after prime system design has begun. Early decisions should favor ET also because that is directed by Army policy. Finally, the specific tasks that the student must perform and specific prime system characteristics are only two of the factors which should affect the media selection decision.

The first step in the development of the guide was to identify the factors to consider when deciding what training to embed. The second was to identify the information needed and formulate the specific questions that must be answered to assess those factors. The third was to structure those questions in a way that would lead the user to a set of logical conclusions, taking into account the changing availability of information during the acquisition process.

To accomplish this, we first reviewed the previous research literature. Eagle Technology, Inc. defined three categories of factors that should affect decisions about what training to embed: Requirements, Opportunities, and Costs.[3] We modified their definitions slightly to produce the following concepts. Requirements-based factors are "high level mission, conceptual, and mission-based factors, and are relatively independent of the prime system" (p. 4-7). Consequently, decisions based on many of these factors can be made relatively early in the acquisition process. Requirements-based factors can influence the prime system design. Opportunity-based factors are derived from the prime system characteristics, the man-machine interface, and the training resources available in the training environment. Cost-based factors include the life-cycle costs of both the prime system and the training system.

Strasel and his associates identified eight major factors related to the probable effectiveness of ET.[10] Those factors are defined in Table 1. We initially added one new factor: policy, which was discussed by Strasel and his associates[10] as a question to be answered ("Are there policy decisions that dictate the use of ET for knowledge and skill acquisition training in the system?" (p. 9)). We established an initial list of sub-factors to consider by combining these two organizational schemes into a 3x9 matrix. For example, the policy factor now had requirements, opportunities, and cost sub-factors.

Following identification and definition of the factors and sub-factors, we reviewed a number of research reports to identify specific questions that others have used in deciding what to embed. Our purpose was twofold. First, the questions suggested changes to our list of factors, either by indicating new factors which needed to be considered, definitions which needed to be revised, factors which were so similar that they could safely be combined, or factors which were not logically sound. Second, the questions suggested how each factor should be considered.

The previously mentioned report by Eagle Technology, Inc.[3] provided our primary source of questions. We also found questions in reports by Strasel[10], Hinton, Braby, Feuge, Stults, Evans, Gibson, and Zaldo[5] and suggestions for questions (not in question form, but readily converted) in Strasel, Dyer, Aldrich and Burroughs.[11] Together, these sources provided a list of 43 questions. We sorted the questions according to the sub-factors we had defined.

TABLE 1. MAJOR FACTORS RELATED TO PROBABLE EFFECTIVENESS OF EMBEDDED TRAINING. (From Strasel, Dyer, Roth, Alderman, and Finley[10])

Factor 1: The Nature of the Tasks and Skills Demanded by the System Concept - What are the Requirements for Sustainment Training.

Factor 2: The Feasibility of Implementation of ET.

Factor 3: Avoidance of ET Interference with Operations.

Factor 4: Need for Training-Specific Hardware Interface Requirements.

Factor 5: System Availability for Training.

Factor 6: Effects on System Reliability, Availability, and Maintainability.

Factor 7: Impacts on System Manpower and Personnel Requirements.

Factor 8: Cost-Effectiveness of ET (compared with alternative sustainment training capable of achieving the same training goals).

We then generated additional questions to fill the gaps. For example, our sources provided no questions about: policy issues; the availability of the prime system for training; Manpower, Personnel, & Training (MPT) requirements and costs; and safety requirements and costs. Our sources also did not distinguish among the various types of ET (fully embedded, appended, and umbilical). We prepared lists of the advantages and disadvantages of each type, and used them as a basis for additional questions.

As we were identifying, generating, and organizing questions, we found it necessary to revise our list of sub-factors. Safety was added. The "Nature of the Tasks and Skills Demanded" was divided into two factors: "Training Content" and "Characteristics of the Training Environment". The definition of the "Need for Training-Specific Hardware Interface Requirements" factor was expanded to include all training-specific interface requirements, not just hardware. Finally, the factor "Avoidance of ET Interference with Operations" was subsumed under the factor "System Availability for Training."

When the question generation process was completed, we had approximately 100 questions. Sample sub-factors, sub-factor definitions, and typical questions are shown in Table 2.

TABLE 2. SAMPLE SUB-FACTORS, DEFINITIONS, AND QUESTIONS

Sub-factor: Policy-requirements

Definition: Conceptual-level statements about the requirements for embedded training. These may range from very general to detailed statements of what is to be accomplished with embedded training.

Sample questions:

Do policy statements or documents indicate an overall preference for, preference against, or neutrality toward embedded training, all other factors being equal?

Are there policy constraints which limit or preclude the use of alternatives to embedded training, such as maneuver areas, live fire ranges, or the use of simulators or devices in the unit?

Sub-factor: System Availability for Training - Opportunity

Definition: The percentage of time during which the prime system can be made available for use as a trainer and still fulfill its prime (combat) mission.

Sample questions:

What percentage of the time can the prime system be made totally available for ET?

Can independent training be provided simultaneously at different duty positions?

Sub-factor: Training Content - Cost

Definition: The life cycle cost of developing, modifying, and maintaining the courseware and other required training materials.

Sample questions:

How complex is the management of the training expected to be? Include management of individual and crew progress, assignment of training sequences, scheduling of training, and scheduling and ordering of all support personnel and materials.

Is extensive networking required in order to provide the ET?

The next step in the process was to review the entire set of questions and sort them into phases on the basis of the expected availability of the information needed to answer each question. The phases were defined as follows:

Phase I: Phase I activities should be conducted about Milestone 0, Concept Studies Approval, for the prime system. The information expected to be available is: general policy and guidance documents regarding both the prime system and its supporting training system; a copy of the Blueprint of the Battlefield; the Mission Need Statement for the prime system; and the expected acquisition schedule for the prime system.

Phase II: Phase II activities are conducted during the Concept Exploration and Definition Phase. The information assumed to be available is (in addition to that available for Phase I): data on the training environment, including the structure of the units expected to receive the prime system, their locations, and the training facilities and resources available to them; and results from the Early Comparability Analysis (ECA).[12]

Phase III: The Phase III analysis should be conducted about Milestone I, Concept Demonstration Approval, of the prime system. The information assumed to be available is (in addition to that obtained for Phases I and II): the prime system Operational Requirements Document; a description of the prime system concept produced by the concept formulation process; detailed information about the predecessor system, if there is one; the results and supporting data of the conduct of HARDMAN Comparability Analysis[9]; and a description of the soldiers who will operate and maintain the prime system.

Phase IV: The Phase IV analysis should be conducted during the Concept Demonstration and Validation Phase of the prime system acquisition cycle. The information assumed to be available is (in addition to that obtained for Phases I, II, and III) data and information from simulations, mock-ups, testbeds, and tests and evaluations.

Next we independently identified the phases at which we expected sufficient information to be available to answer each question. We then compared our results, resolved differences, and assigned each question to one or more phases.

For each Phase, the questions were organized into a logical sequence leading to training alternative recommendations. Many complex questions were divided into a series of simpler questions. Flow diagrams were developed. Finally, textual explanations of each flowchart segment or block, and worksheets to present the results, were developed.

Questions about the costs of training alternatives were grouped separately into a Training Alternatives Cost Summary (TACS). The TACS could be completed at any time, but usually following the Phase III or Phase IV analysis. Cost questions were organized into a TACS Worksheet, rather than a series of flowcharts.

RESULTS: A GUIDE FOR EARLY EMBEDDED TRAINING DECISIONS

These procedures produced A Guide For Early Embedded Training Decisions,[13] which consists of nine sections and an appendix. Sections 1, 2, and 3 provide introductory material and "how to use" information. Sections 4, 5, and 6 consist of flowcharts for phases I, II, and III/IV, respectively. Within each phase, flowcharts are separated into blocks of related questions. Each block includes questions to be answered by the evaluator and is accompanied by help text that explains the decision process represented in that block. ET decisions are made on the basis of how the evaluator answers the flowcharted questions. Section 7, the Training Alternative Cost Summary, requires the completion of a cost estimating worksheet, rather than working through flowcharts, as required in the other phases. Appendix A provides information regarding the ten factors listed in Table 2 of this paper.

USER FEATURES

The ET guidelines were developed to provide specific early guidance to the user in making decisions about embedded training. To this end every effort has been made to keep the procedures performed by the user as simple as possible. Basically, the user is required to step through a series of flowcharts. Each flowchart question constitutes a decision point, where a "YES" or "NO" answer leads to another question, and so on, until a decision is reached. Figures 1 and 2 are examples of Phase II flowcharts for Blocks 2 and 3, respectively. Help text is provided with each flowchart block to explain the purpose of that block of questions and to provide the logic and rationale behind the selection and sequencing of the flowchart questions.

For keeping records of the decisions made, the Guidelines include a Training Alternative Summary Matrix Worksheet. Figure 3 is a completed sample matrix showing the results of a Phase II analysis of four prime system functions: navigation, vehicle maneuvering, target acquisition, and weapons function management. Training alternatives are identified as Preferred, Recommended, Alternative, or Excluded depending on whether they best satisfy, fully satisfy, minimally satisfy, or fail to satisfy training requirements.

The guidelines also provide worksheets to help the user estimate the costs of ET and other training system alternatives. The Training Alternative Cost Summary may be used to compare alternative training systems on four cost categories: Design and Development, Procurement, Maintenance, and Operations. The cost worksheets supplement the decision flowcharts by providing additional criteria for making ET decisions. A sample worksheet used in estimating Design and Development costs is included as Figure 4.

Phase II, Block 2. Can the prime system support ET, given MPT and RAM requirements?

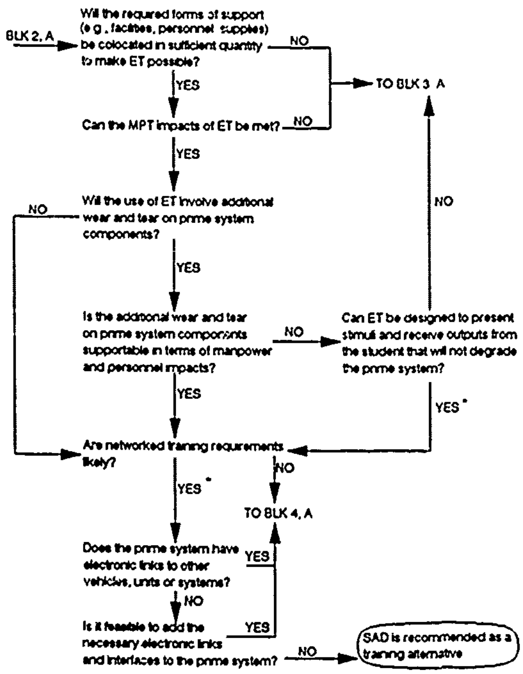


Figure 1. Block 2 Flowchart for Phase II Training Decisions.

Phase II, Block 3. Are other training alternatives supportable in terms of MPT and training facility requirements?

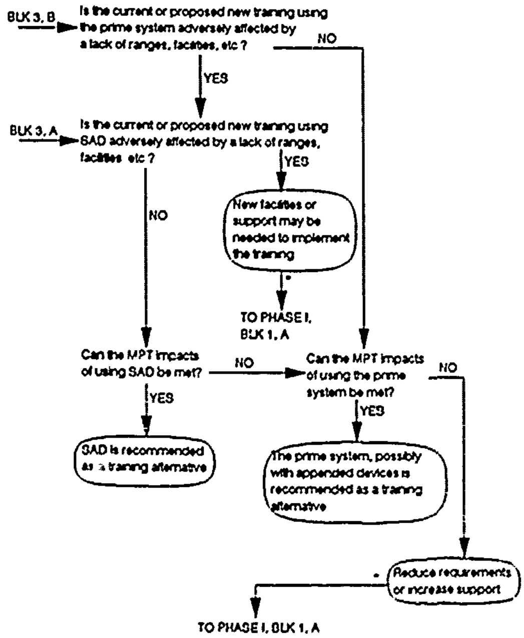


Figure 2. Block 3 Flowchart for Phase II Training Decisions.

TRAINING ALTERNATIVE SUMMARY MATRIX

MISSION, FUNCTION, TASK OR SUBTASK	AET	ET			APPENDED DEVICE	SAD	CBI	CLASSROOM
		FULLY	APPENDED	UMBILICAL				
TARGET ACQUISITION		R	R					
NAVIGATION							P	P
VEHICLE MANEUVERING						R		
WEAPONS FUNCTION MANAGEMENT		R	R	R				

LEGEND
P = PREFERRED
R = RECOMMENDED
A = ALTERNATIVE
E = EXCLUDED

Figure 3. Sample Training Alternative Summary Matrix

STATUS AND PLANNED ACTIVITIES

The ET guidelines have been refined and improved based on comments provided by experts in the embedded training area, but the guidelines have not yet been applied to a system acquisition. We plan to apply the guidelines to improve the quality of ET decisions for the Armored System Modernization (ASM) Program. The application will occur in conjunction with the development of the integrated training system for the ASM tank variant. We expect that some changes to the guidelines will occur as the direct result of this practical application.

Currently the user of the ET Guidelines must work through the flowcharts and associated worksheets manually, recording decisions and recommendations on training alternative and cost summary worksheets. However,the flowcharts and help sections were designed with a computer-based implementation in mind. One advantage of computer-based ET Guidelines is the increased speed with which the user could render decisions about the advisability of using ET for the various missions, functions or tasks. Another advantage is that the computer could keep track of the decisions made as the user progresses through the flowcharts, alleviating the user from the tedious task keeping detailed records of the decision process and maintaining a permanent audit trail of the decision process. Such an audit trail could greatly facilitate subsequent review and revision as the prime system and the training system evolve. The computer could also keep track of media decisions and maintain a file of decisions and recommendations that affect the cost of the training alternatives. A computer implementation of the ET guidelines is planned following their application to the ASM program, if funds are available for that implementation.

	AET	APPENDED	EMBEDDED TRAINING			SAD
			FULLY	APPENDED	UMBILICAL	
Design & Development. What is the cost of designing and developing the training subsystem for each training alternative? Consider the following:	$____	$____	$____	$____	$____	$____
What is the cost of designing new (or upgraded) ranges and facilities?	$____	$____	$____	$____	$____	$____
What is the development cost of the training management system? Consider how complex the management of the training is expected to be. Include management of individual and crew progress, assignment of training sequences, scheduling of training, and scheduling and ordering of all support personnel and materials.	$____	$____	$____	$____	$____	$____
What are the costs of developing supporting documentation (e.g., Instructor/Operator manual, maintenance manuals, etc.)?	$____	$____	$____	$____	$____	$____
What are the courseware development costs?	$____	$____	$____	$____	$____	$____
Does the training alternative require the development of complex simulations? If so, do these simulations require a direct view of the outside world?						
Is the courseware development required within the "state of the art"?						
Does the training require that the simulations function in an interconnected network?						
Must the hardware and software interact with system components that provide simulated motion (e.g., a motion platform)?						
What are the hardware and software development costs?	$____	$____	$____	$____	$____	$____
Does the training require that the simulations function in an interconnected network?						
Must the hardware and software interact with system components that provide simulated motion (e.g., a motion platform)?						
Is training system component ruggedization required?						
Is training system component miniaturization required?						

Figure 4. Training Alternative Cost Summary

SUMMARY

Problems in implementing embedded training have prevented it from realizing its full potential. These problems are the result of the requirement to specify embedded training requirements well before the information traditionally used in making training media decisions (e.g., task characteristics) is available. Previous ET work has not been successful in providing specific procedures for making early ET decisions, but some researchers[3,5,10,11] in the area have provided the raw materials (i.e., the concepts and questions) for making these decisions. Starting with known characteristics of effective embedded training systems, a bias for ET derived from its recognized advantages and the assumption that the decision process must be phased and linked to information availability, a set of guidelines were developed for making early embedded training decisions. The development process entailed identifying approximately 100 questions and organizing these by categories for inclusion in media decision flowcharts and cost summary worksheets. The guideline procedures, require the analyst to manually work through a series of detailed decision flowcharts and training alternative cost summary worksheets to produce early embedded training recommendations. A computer-based version of the ET guidelines is planned as is their application to the Armored System Modernization training system acquisition.

REFERENCES

1. Breglia, D.R. (1985). Design considerations for on-board simulation. Proceedings of the IEEE 1985 National Aerospace and Electronics Conference (NAECON 85).

2. Department of the Army (1987, March). Policy and guidance letter, Subject: Embedded training. Washington, D.C., Author.

3. Eagle technology, Inc. (1988, March). Early Training Resource Estimation Model (ETREM). Winter Park, FL: Author.

4. Finley, D.L., Alderman, I.N., Peckham, D.S., & Strasel, H.C. (1988). Implementing embedded training (ET): Volume 1 of 10: Overview (Research

Product 88-12). Alexandria, VA: U.S. Army Research Institute for the Behavioral and Social Sciences.

5. Hinton, W.M., Braby, R., Feuge, R.L., Stults, A.H., Evans, S.M., Gibson, M.R., & Zaldo, W.T. (1990). A design architecture for an integrated training system decision support system (Research Report 1566). Alexandria, VA: U.S. Army Research Institute for the Behavioral and Social Sciences.

6. Hoehn, A.J., Woolman, M. & Glaser, R. (1969). Operational context training in individual technical skills (HumRRO Professional Paper 35-69). Alexandria, VA: Human Resources Research Organization.

7. Hoskins, B.J., Jorgensen, W.F., Manglass, D.A., & Reynolds, R.E. (1989). Lessons learned from currently fielded Navy embedded training systems (Naval Training Systems Center Technical Report 89-011). Orlando, FL: Naval Training Systems Center Human Factors Division.

8. Hughes, R.G. (1986). Aircrew embedded training: A logical extension of current and emerging technologies. Proceedings of the IEEE 1986 National Aerospace and Electronics Conference (NAECON 86).

9. Parrish, D.L. (1990, May/June). MANPRINT methodologies: Early Comparability Analysis and Hardware vs. Manpower Comparability Analysis. MANPRINT Bulletin, IV (6).

10. Strasel, H.C., Dyer, F.N., Roth, J.T., Alderman, I.N., & Finley, D.L. (1988). Implementing embedded training (ET): Volume 2 of 10: Embedded training as a system alternative (Research Product 88-22). Alexandria, VA: U.S. Army Research Institute for the Behavioral and Social Sciences.

11. Strasel, H.C., Dyer, F.N., Aldrich, R.E., & Burroughs, S.L. (1988). Review of eight Army systems: Characteristics and implications for embedded training (Research Note 88-14). Alexandria, VA: U.S. Army Research Institute for the Behavioral and Social Sciences.

12. U.S. Army Personnel Integration Command - Soldier Support Center (1987). Early Comparability Analysis (ECA) procedural guide. Alexandria, VA: Author.

13. Witmer, B.G. & Knerr, B.W. (1991) A guide for early embedded training decisions. (Research Product 91-14). Alexandria, VA: U.S. Army Research Institute for the Behavioral and Social Sciences.

ABOUT THE AUTHORS

Dr. Bob Witmer is a Research Psychologist currently assigned to the Training Systems Development Team at the U.S. Army Research Institute PM TRADE Field Unit, Orlando, FL. His research activities during his eleven years with the Institute include the design of job aids for training developers and evaluators, the measurement of soldier performance under simulated future battlefield conditions, and the development of performance measures and feedback formats for Army training systems. Dr. Witmer earned a Bachelor of Science degree in Electrical Engineering from Tennessee Technological University and a Ph.D. in Experimental Psychology from Georgia Tech.

Dr. Bruce W. Knerr is a Research Psychologist and Leader of the Training Systems Development Team at the U.S. Army Research Institute PM TRADE Field Unit, Orlando, FL. He has over sixteen years experience conducting and directing research and development in Army training. His primary areas of research have been development and application of training technology and development of early Manpower, Personnel, and Training requirements estimation methodologies. He earned his Ph.D. in Industrial Psychology from the University of Maryland.

APPLICATION OF A

KNOWLEDGE COMPILATION MODELOF INSTRUCTION TO EMBEDDED TRAINING

Thomas F. Carolan
University of Central Florida • Institute for Simulation and Training • Orlando, Florida

Kent E. Williams
Virginia Polytechnic Institute and State University • Management Systems Laboratories • Blacksburg, Virginia

Nina Chatham
ORAU Postgraduate Research • Participation Program • Naval Training Systems Center • Orlando, Florida

ABSTRACT

Considerable research has been directed at understanding the processes involved in acquiring and using knowledge and skills. One focus of this cognitive research is the application of formal models of learning and knowledge representation to the design of computer based instruction. Advances in adaptive instruction and intelligent tutoring have been driven by implementing explicit models of the knowledge to be learned, and the strategies used to communicate that knowledge. Two recent experiments, employing Navy personnel, have demonstrated the effectiveness of using a formal approach to instruction in an embedded training environment.

The design of the instructional material began with a fine grain analysis to determine the knowledge to be learned and to develop the basic structures upon which the adaptive processes operate. There is evidence that curriculum content which is generated from the results of an explicit cognitive analysis process enhances learning. In the first experiment the effectiveness of using a cognitive analysis process to structure the information and an adaptive process to sequence the information was evaluated for domain naive students. In the second experiment the effectiveness of the knowledge compilation approach was evaluated using Navy instructors. The results of this research are discussed in the context of the application of current cognitive learning research to embedded training.

INTRODUCTION

During the past decade, the cognitive psychology research effort has advanced the development of instructional theory through the application of cognitive models of competent performance to the development of more precise instructional methods and strategies (Glaser & Bassok, 1989). Over this same time period, the study of intelligent tutoring systems (ITS) has emerged as an interdisciplinary research field (Johnson, 1991) in which artificial intelligence techniques are applied to implement these cognitive-based methods and strategies as instructional system programs and architectures. An emphasis on the implementation of more explicit instruc tional methods and strategies should also benefit more tradi tional computer based training systems.

The military services have moved to provide effective embedded training (ET) through building training capabilities into or adding them onto operational systems. Williams and Reynolds (1989), in a review of existing ET systems, pro posed that the implementation of cognitive learning principles and intelligent tutoring strategies would improve the instruc tional technology component of ET and lead to considerable training gain over conventional computer based instruction.

What types of instructional strategies will be most effec tive for embedded training environments? On board embed ded training (ET) provides console operators with a high fidelity environment in which to practice, refresh, and refine the highly perishable cognitive and motor skills needed to skillfully interact with complex systems. Also, ET provides the opportunity to gain the new knowledge and skills re- quired for qualification on advanced watch stations. The objectives of an ET session is to build on existing knowledge, to diagnose and correct deficiencies as efficiently as possible, and to allow for consolidation of skills through practice. The training effectiveness of ET sessions will depend upon instructional technologies and strategies that promote effi- cient acquisition and retention of skills and knowledge.

Candidate instructional technologies were selected to be implemented, and experimental evaluations of the impact of these technologies on the acquisition and retention of skills and knowledge were initiated. Two of these experimental evaluations were conducted on the Navy's Lesson Translator (L-TRAN) embedded training system for the general purpose tactical consoles supporting Navy Tactical Data System (NTDS) sensor and weapons operations. The first experiment (Williams et al., 1989) evaluated, for entry level console operators, the impact on training gain of structuring the lesson content based on a cognitive analysis process, and of adapting the sequence of training to individual performance. Both strategies were found to have an effect on learning with

the domain naive students. A second experiment extended that effectiveness evaluation to more experienced users, and identified the distribution of practice throughout the lesson as a contributing factor of the increased effectiveness. This paper summarizes the research, discusses the application of a cognitive analysis process to the design and sequencing of instructional exercises, and interprets the research results within a knowledge compilation framework (Anderson, 1983).

PROCEDURAL LEARNING AND INSTRUCTION

Tactical console operations can be characterized as involving the skillful application of specific perceptual, cognitive, and motor operations to the goal-directed processing of complex information patterns. Developing competence at organizing appropriate sequences of operations in response to particular conditional contexts can be understood as a process of acquiring and proceduralizing knowledge. Given this perspective, it was determined that an appropriate instructional strategy for embedded training of console operations could be based on applying a cognitive skill acquisition approach to the problem. One assumption regarding the implementation of this approach is that the essential information and operational procedures required for performing a task, such as those involved in operating a tactical console, constitute the elements of an ideal model of competent performance (Anderson, Boyle, Corbett & Lewis 1990). For the operator, learning to perform a task can also be interpreted as an active knowledge construction process through which a representation, or cognitive model, of the essential information and required procedures is developed. This model then serves as a basis for the operator's future interaction with the device.

Generating Rules

There is considerable research evidence to support a characterization of human processing as inclined towards encoding patterns of information in a manner consistent with the generation of rules. People are very good at abstracting rules from examples, when those examples involve concrete events to which they can relate (e.g., Cheng, et. al., 1986). One difference between experts' and novices' knowledge is that experts tend to functionally organize information in cause and effect relations that are linked to the conditions under which it can be used (Glaser & Bassok, 1989). The utility of coding knowledge in terms of rules is further illustrated by examples from instructional research. For example, Chi, Bassok, Lewis, Reimann and Glaser (1989) demonstrated that good students are able to explain the conditions and consequences associated with a specific action, whereas poor students are not. Good students also generate more complete condition-action rules during the learning process. They construct such rules from the instructional material presented to them (Bovair & Kieras, 1990).

The production system framework (Newell & Simon, 1972, Anderson, 1983) for modeling cognitive processes provides an explicit formalization of cognitive skill development. Models implemented as production systems form the basis of many current intelligent tutoring systems. Briefly, a production system consists of a set or network of production rules or condition-action pairs of the IF-THEN form, a context or environment, and a control strategy. The condition side of the production rule specifies task goals and subgoals, along with specific states of the environment. The conditions are the inputs to the rule. If all of the conditions of a rule are matched, then the rule is triggered and the action side is executed. Essentially, production system models attempt to establish links between cognition and the production of actions.

Knowledge of procedures can be represented by sets of production rules. These sets of productions are directed towards achieving goals and incorporate goals in their structure, starting with a high level goal and decomposing it into subgoals. The goal structure, in turn, assists in specifying an appropriate set of production rules. The production rules can be reorganized depending on the goal structure of the task. Anderson (1986) has noted that by establishing a goal tree, one can predict what rules are likely to be composed by students.

From this perspective, the console operator's knowledge and skill is structured as a production system model of the instructional information. The goal of training is to provide an environment for students to acquire or refresh these production rules (Anderson, Conrad, & Corbett, 1989) which represent the tasks to be performed and the information necessary for their performance.

Knowledge Compilation

The learning process through which procedural knowledge is acquired has been referred to as knowledge compilation (Anderson, 1983, 1986). An overview of this approach with its implications for instruction provides a basis for understanding the acquisition of console operation skills. Knowledge is thought to be encoded as declarative structures through language comprehension and perceptual processes. However, in order to convert this knowledge of facts (declarative knowledge) into knowledge of how to correctly use those facts to produce actions in a particular situation (procedural knowledge), the declarative knowledge must be applied and interpreted in the context in which it is to be used.

Declarative knowledge is typically applied in a trial and error fashion resulting in the generation of more efficient and task specific production rules. Performance becomes more goal directed. As practice continues, the knowledge becomes more procedural, the need to refer to declarative knowledge is reduced, and the application of the rules becomes more automatic. This transition from using declarative knowledge to using procedural knowledge has been called proceduralization (Anderson, 1983). Through knowledge compilation, high level rules that link key information and action components of the situation in a more efficient manner are produced. Small steps are "composed" into single units which can be accessed independently and applied when appropriate. In addition to composing rules into larger 'chunks' and proceduralizing successful rules, new information about specific conditions and actions may be added to declarative memory With continued practice, more productions can be formed based on these additional declarative structures. Anderson (1986) and others, have found that experts have difficulty recalling procedures which they used early in their training As experts acquire knowledge, old procedures are composed to form higher level condition

action segments. At some point, it may be difficult for experts to retrieve the lower-level procedures from which the higher-level procedures were formed.

In addition to the compilation process, empirical evidence exists to support a strengthening process (Anderson, 1982, Anderson, et al., 1990). As declarative and procedural knowledge are used, they are applied more efficiently, allowing resources to be used to process new information. Anderson reports evidence which demonstrates that encoding time is reduced from first to second application of a production. This is interpreted as the compilation of knowledge into procedural form followed by a gradual reduction in encoding time as the production rules become strengthened. From a knowledge compilation perspective, learning involves a process of acquiring new declarative knowledge using existing productions, applying this declarative knowledge to new situations, compiling task-specific productions, and strengthening both declarative and procedural knowledge.

A number of investigators (Anderson, et al. 1990; Carroll, 1990) have identified a problem with technical instructional material. Typically, some of the information the trainee needs for task performance is omitted. The student must then infer the missing information or discover it through trial and error. Also, information that is not useful for either understanding or performance in the task domain is presented. The student then uses time and cognitive resources attempting to interpret this information in the context of the task environment (Kieras, 1987a). When instruction is not precise and to the point, the cost in terms of time and cognitive resources increases, as does the potential to negatively affect performance through incorrect interpretation (Anderson & Jeffries, 1985). A number of researchers (Kieras, 1987a; Reder, Charney & Morgan, 1986) have demonstrated the effectiveness of concise and well focused textual instruction.

The cognitive modeling perspective provides a framework for determining the content of instruction. A cognitive model, or in Anderson's terminology, an ideal student model, provides a cognitive level representation of training requirements for achieving competent performance. One instructional strategy is to embed this ideal model into the instructional system and use it to diagnose student errors and generate instruction. Evaluations of these intelligent tutoring systems indicate that they can achieve better results than standard classroom instruction (Anderson et al. 1990). Anderson also reports that instructional materials designed to communicate the information in the ideal model can be more effective than standard texts even without an adaptive tutoring component.

The explicit cognitive model framework has generated a number of instructional principles that provide guidance with reference to designing and presenting material to the trainee in a way that will make learning more efficient. For example, learning will progress more efficiently when the individual can explicitly observe the rules to be learned. Therefore, by keeping the language concise and focused on an explicit encoding of the facts, relations, and rules, the process of interpreting declarative knowledge can be made more effective. In addition, graphical representations that make logical and spatial relationships explicit will reduce the inference requirements and facilitate interpretation (Larkin & Simon, 1987). When the focus of declarative instruction is skill development, learning can be enhanced by presenting material in a format appropriate for mapping onto the goal and subgoal structure of the task environment. The knowledge representation should specify not only the rules but the functions of these rules, any preconditions that apply, and the anticipated consequences.

The knowledge compilation model makes certain assumptions about how the student's knowledge changes as the student progresses through a learning program. These assumptions can be used to guide a strategy for presenting the components of instruction. For example, the components of a rule should be explicitly identified and coupled to allow for more efficient composition of rules. This will minimize the working memory load required during the learning process, and ultimately during performance.

Proceduralization can be facilitated by providing interactive practice opportunities and feedback to assure that trainees have correctly interpreted the declarative instructions. The knowledge and skill objectives of the instruction, the structure of the domain to be learned, and the existing skill level of the trainee set the bounds on the appropriate level of composition to be achieved. Within these bounds, rules can be composed into higher level productions or decomposed into substeps By providing opportunities for practice at both the composed and the substep levels, the trainee, or the system, can adapt instruction and apply practice based on the strength of existing declarative and procedural knowledge.

THE INSTRUCTIONAL METHODOLOGY

Console operators typically monitor or manually perform activities which include: searching for, detecting, and tracking targets with radar/sonar, identifying the target track(s) as friend or foe, evaluating and ranking by threat posed (e.g., platform, proximity, speed, heading), assigning and engaging weapons to counter targets, and assessing the results of engagement. These tasks are complex and require a significant amount of knowledge as well as cognitive and psychomotor skills. Operators must possess the required knowledge so that they do not have to look up information and procedures; they must also possess sufficient speed to operate their console while handling multiple incoming threats.

The development of a production system model of the instructional material is accomplished by conducting a cognitive task analysis (Kieras, 1987a, 1987b, 1988; Williams, Reynolds, & Carolan, 1990). The cognitive analysis methodology shares many of the principles and processes of the GOMS model of Card, Moran and Newell (1983). The purpose of the cognitive task analysis is to determine what specific knowledge has to be learned through the training and to develop representations of that knowledge which are consistent with those produced by mechanisms of human learning, such as production memory and its associated cognitive processes. The technique of generating production units for instructional systems has been applied to mathematics, (Anderson, 1981, Reif, 1989), programming, (Anderson, Conrad & Corbett, 1989), and tactical console operations (Williams, et al., 1989).

Cognitive Task Analysis

Consistent with the approach discussed by Kieras (1987a, 1987b), the initial analysis focused on the development of a hierarchy of task goals from the information required to effectively operate a tactical console. This analysis involved

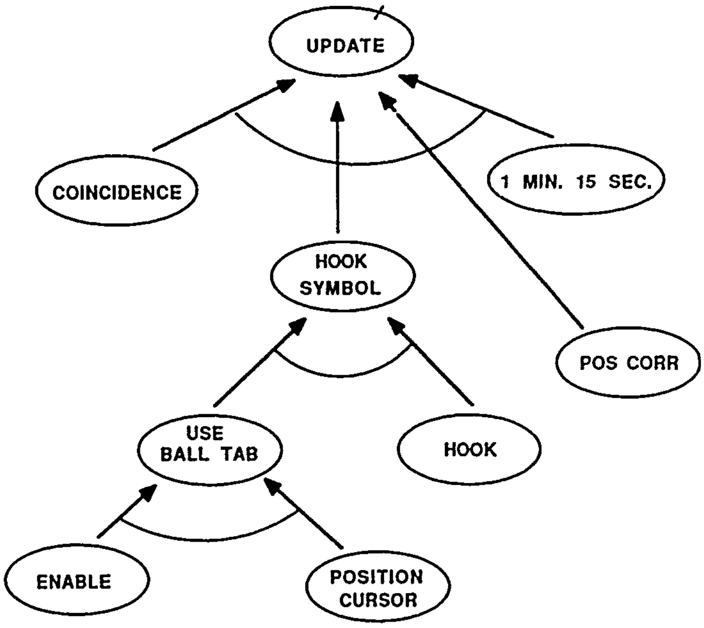


Figure 1. A partial graph of a method to accomplish the UPDATE goal.

the specification of high level goals, and the identification of the tasks that could be performed on the device which directly related to the attainment of those goals. For console operations, this analysis provided information at different levels. (1) a description of the physical layout of the console with locations of controls and outputs; (2) a description of the various functions of the console; (3) the external console operating procedures; (4) the functional relationships between console parameters and task parameters; and (5) how to use the console parameters to perform the relevant tasks.

A device description hierarchy (see Williams, Reynolds, Carolan, 1989; 1990 for detail) consisted of the information that would be used to explain or describe the relationships among the input and output components of the console. This hierarchy included things like action buttons and display components. A goal or function hierarchy consisted of the information that explains what tasks the console is to be used for. It is used for accomplishing such tasks as acquiring a new track on a potential target, updating information on a firm track, and identifying characteristics of the target. Each element of the function hierarchy is a goal that the user or trainee must accomplish. Achieving each goal requires using elements of the device description hierarchy to perform specific tasks. By linking the elements of the device descrip tion hierarchy to the appropriate elements of the function or goal hierarchy, a task goal hierarchy is developed.

Consistent with the GOMS approach, once the task goals are identified, the methods which make up the task proce dures are detailed. Each method specifies a goal or subgoal to be accomplished, as well as the sequence of steps to be executed, and conditions to be met, in order to accomplish that task goal. All responses that the individual requires in performing the to be learned tasks are specified and associ ated with each task goal and subgoal in the hierarchy. All conditions associated with each individual response are identified and linked to that response. All constraints required by console operating procedures are identified and linked to the appropriate responses. Each step of each method consists of an operator or action which is executed. Operators can be perceptual, such as observing the location of a symbol; actual, such as pressing a button, or cognitive, such as making a decision, or storing or retrieving memory information. This detailed cognitive analysis of the domain knowledge then organizes the goals, subgoals, methods, and operators into a graph hierarchy. A portion of the task-goal hierarchy representing tactical console operations in AND/OR graph form is illustrated in Figure 1.

Goals and subgoals can be nested within methods and submethods when specifying the operation of a device, and within a method, other methods may be called by a particular step. If more than one method may be used, the GOMS model requires a selection rule which discriminates between alternative methods, and determines which specific method should be used. For example, an alternative method for hooking a target is to press the sequence button and then the hook button. This method will sequence control to the next target. A selection rule would specify the conditions under which each method is used.

The result of this analysis is the detailed specification of all of the knowledge needed to complete tasks associated with console operations. The specification of methods within the GOMS model essentially identifies a set of production units and rules for relating these productions through the goals and subgoals. The goal and subgoal analysis divides the problem into small enough parts so that the methods or rules can be generated. The specification of these rules is the development of the production system that simulates system functions and operations. Exercises developed as a result of this cognitive analysis process explicitly specify the cognitive model which the trainee must acquire to effectively produce

correct responses consistent with the domain knowledge. The goal of the training program is achieved when the trainee works through the exercises and develops an accurate model of the knowledge domain.

Developing Instructional Content

Instructional frames were developed from this hierarchical production system model. When working with existing lessons, the lesson content is restructured based on the production system model developed through the cognitive task analysis. Each individual rule or method in the lesson, as well as each rule that combines relevant facts and/or lower level rules, is composed into an individual exercise. An exercise can consist of information presented on a single screen or frame, or over a sequence of frames. A lesson is made up of a number of exercises each presented over one or more frames. This breakdown of the lesson into a hierarchy of individual exercises is not inconsistent with the modular structure of the L-TRAN system. Each subject matter module in a lesson is considered to be an independent unit in content and in presentation order. These subject matter modules represent the task goal level of the GOMS hierarchy. Our explicit use of the GOMS methodology separates the material within a subject matter module into instructional units at the methods and at the operator levels. It is these units which we refer to as exercises. Each exercise explicitly describes the subgoal or goal state which triggers the production represented by the frames making up the exercise. In working through the frames the student learns all the declarative facts and how to compose or compile these facts so that when specific cues or subgoals are set, specific actions or sequences of actions are triggered. Each exercise frame created with this methodology is explicitly linked to the conditions, actions, or other rules that make up a production. This methodology is consistent with the principles for lesson design discussed in the L-TRAN style guide, but provides for a more specific design process.

Each exercise consisted of three parts: (1) an exposition of the information specifying a rule; that is, the knowledge to be learned; (2) a problem or example which required a response to a question or the performance of a procedure, and which tests the student's ability to use the knowledge and provides practice in implementing the procedure, and (3) a set of diagnostics that determine the source of any error(s) in terms of the specific conditions or actions of a production for which the student's knowledge is weak.

Exercise Sequence

Advantages of coupling instructional content to a rule-based cognitive model include efficient delivery of exercise content and effective diagnosis of errors relating to specific rules, declarative facts, or combinations of rules. Since each exercise created employing this methodology can be explicitly linked to pieces of a production in the form of conditions, actions, or other rules, the training program can query the trainee to determine what conditions, action, or other rules have been or not been learned. This assessment of student strengths and weaknesses can then help to determine what lesson content should be presented next.

The hierarchical structure of exercises, the specification of goals and subgoals, and the derivation of exercises from underlying productions are all consistent with empirical evidence from cognitive learning experiments. The structuring of the information content of exercises is consistent with the way procedural knowledge is organized. The problem segment of each exercise provides the student with the opportunity to practice (Rosenbloom & Newell, 1986) what is learned from the exposition segment, and thereby use declarative knowledge to develop procedural structures. By using diagnostics to localize strengths and weaknesses, more specific feedback can be provided (Hayes-Roth, Klahr, & Mostow, 1981) to the student and the system can be guided in its selection of the next best exercise for that student.

The capability to select an exercise which overlaps most with what the student knows can be built into an instructional system through an adaptive selection heuristic. The adaptive heuristic facilitates learning by sequencing exercises so that the contents of any two consecutive exercises share as much knowledge as possible. The sequence is based upon what each individual has learned, and therefore the sequence can be different for any two individuals. Upon failure or success on a particular exercise, the system searches for and selects an exercise for presentation which overlaps most with what the student knows and least with what the student does not know. This method exploits strong existing knowledge to build new knowledge. To keep track of the student's progress, the system keeps a record called the student model. The student model records the status of each criterion frame and diagnostic as the lesson progresses. The strength of each piece of knowledge or production rule is determined by the strength of the individual elements which make up that knowledge. The student completes the criterion frame, and if unsuccessful, the system presents the appropriate diagnostics. Then the strength of all rules is updated to reflect the current state of the student's knowledge of the lesson. The heuristic search routine selects the next frame to be presented, based on the information about the current strength of the individual knowledge and rule units which make up the lesson (see Williams, Reynolds, & Carolan, 1990 for detail).

In situations where adaptive exercise sequencing is not possible, the knowledge compilation and strengthening processes suggest guidelines for fixed exercise sequences and branching. When lower levels in the hierarchy of related facts and rules are strengthened initially, with ample opportunity to interact and develop procedural representations, the effect spreads to other related rules. Composite rules, further up in the hierarchy, are also strengthened by virtue of their microunits, thereby facilitating their assimilation.

EXPERIMENT ONE

In the first experiment (Williams et al., 1989), a study was conducted to evaluate the effectiveness of lessons which were restructured based on the cognitive analysis process and sequenced using the adaptive methodology. The domain was tactical console operation procedures. An intelligent computer aided instruction system composed of cognitively developed instructional content and the adaptive instructional strategy was integrated into the Navy's L-TRAN system. A detailed discussion of this experiment was presented at this conference last year (Williams, Reynolds, Carolan, 1990) and will be summarized below.

Forty-eight enlisted Navy students, none of whom had any prior experience with tactical training consoles, were

randomly presented one of three versions of a basic lesson on a standard software emulation of an Navy Tactical Decision System (NTDS) console. The content of the L-TRAN lesson was restructured based on the cognitive task analysis process. An adaptive exercise sequencing heuristic was implemented in software, and diagnostics were developed. The result was three versions of the L-TRAN lesson: (1) the original version currently in use, (2) a cognitive version in which the content was structured according to production rules developed from a cognitive task analysis and presented in a fixed sequence, and (3) an adaptive version in which cognitively structured exercises were sequenced, based on the individual's performance.

There was a dramatic impact on training gain when the instructional content of the test lesson was restructured using the cognitive engineering process. Trainees using the cognitively restructured lesson made 65% fewer errors on a final performance test than the control group (See Figure 2). This final performance test required the trainee to interact with the device to recognize the appropriate task goal to be achieved for a given situation and to perform the explicit procedures required for the achievement of that task goal. These results confirmed, for the domain of console operations, that structuring the information to be learned according to guidelines developed from a production system model of knowledge representation can have a profound impact on learning effectiveness.

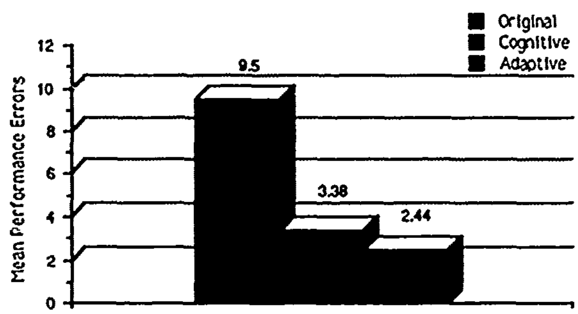


Figure 2. Mean number of errors in each group on the final performance test trial.

In this experiment, performance was further improved when the sequence in which exercises were presented was controlled by an adaptive selection heuristic. After an initial learning trial, the use of this adaptive selection heuristic resulted in increased efficiency of learning; that is, more improvement in learning per unit of learning time. These results indicate that by taking advantage of each individual's prior knowledge and history of performance on the lesson, adaptive frame sequencing is a means for improving the efficiency of ET sessions. To the extent that students with varying degrees of competency are using the ET, the effectiveness and efficiency of adaptive exercise sequencing should be maximized.

The results of this experiment provided convincing evidence that, for naive trainees with little existing domain specific declarative knowledge and virtually no domain specific procedural knowledge, the more formal cognitive approach to instructional design was more effective than a traditional, less formal approach. The target population for embedded training may consist of fleet personnel with entry level to advanced training and experience backgrounds. The trainees using embedded training lessons enter a particular lesson or series of lessons with individual differences in both the declarative knowledge they bring to the lesson and in the device specific procedural knowledge. Will exercises which have been structured based on the cognitive modeling framework make any difference when the trainees are already competent in both the general knowledge domain, and in console operation procedures?

EXPERIMENT TWO

A second experiment (Carolan, Williams, Moskal, 1991) was designed to: 1) transition the previous research on an NTDS console emulator to research on NTDS operational consoles, 2) evaluate the effectiveness of this knowledge engineering methodology with more experienced subjects, and 3) evaluate the impact on learning gain of varying the opportunities for performance feedback.

In the L-TRAN embedded training research described above, fully adaptive sequencing was possible only when NTDS console emulator systems were used. The emulator system allowed for modifications to the control software to implement the student model and sequencing algorithms. In this second experiment, NTDS instructional consoles were used and modification of the NTDS software was not feasible. For this reason, it was not possible to include further evaluation of adaptive exercise sequencing in the second experiment.

An L-TRAN lesson designed to teach the material and procedures required for an advanced watch station was subjected to the cognitive analysis process described in the previous sections and detailed in Williams, Reynolds, and Carolan (1989). The content of the lesson was restructured based on this analysis. Each exercise in the lesson was tightly constrained to match a single rule, or composed rule. Not every rule however, was represented as an exercise in the lesson. Since this was an advanced lesson, the production rules included in the lesson represented middle to upper levels of the skill. Subgoals which involved lower level productions were not always explicitly presented. These include recognition of basic symbology, location of fixed and variable action buttons, composition of low level methods such as hook and ball tab. These productions are prerequisite to the current lesson, and should already be well-learned.

Each exercise generally consisted of two parts: (1) expository information specifying the knowledge to be learned; and (2) a problem or example which required a response or the performance of a procedure. The second part of the exercise required the student to practice using the knowledge and provided an opportunity to implement a proceduralization process and to receive feedback. Three versions of the lesson were compared. These versions differed in the structure of the expository and the practice parts of the lesson. For one version, the expository information was structured to be explicitly related to the sequence of rules developed from the cognitive analysis process described above. The lesson proceeded, for the most part, in a fixed sequence in which expository text explicitly defining a rule is presented and is followed by practice in order to test one's interpretation of the text and to incorporate the rule as a procedure. This sequence of exposition and practice continued at each step in the lesson. For example, referring back to

Figure 1, each node in the graph of UPDATE represents a rule, and each rule is represented in the lesson by an exercise that includes both an expository and practice component. This version of the lesson is consistent with the fixed sequence cognitive version of the previous study (Williams et al. 1989), and is therefore referred to as the cognitive version. A second version of the lesson was developed using the same restructured information as in the cognitive version. In this version, however, the student did not have the opportunity to evaluate one's interpretation of the declarative text information, and receive appropriate feedback, at each of the lower level nodes. The only opportunities for practice occurred at the level of the composite task goals (i.e., the UPDATE node in Figure 1). Hence, we refer to this version of the lesson as the composite practice version. The remaining version is the original L-TRAN lesson in which the structure is not based on an explicit cognitive analysis, and, as in the composite practice version, the opportunity to practice is usually provided only after all the required steps have been presented.

In summary, the result was three versions of the lesson: (1) the original L-TRAN version currently in use (original); (2) the cognitive version in which both the declarative and procedural components were cognitively restructured (cognitive); and (3) a composite practice version in which only the presentation of declarative content was restructured (comp. practice). The experiment was designed to test the effectiveness of the cognitive and composite practice versions as compared to the original version.

Thirty-six enlisted Navy instructors participated as trainees in this experiment. The rank and rate of the instructors varied. They were selected from two categories based on their NTDS console experience. The less experienced participants were not NTDS console instructors and had little or no NTDS console experience. The more experienced participants were NTDS console instructors, but were not experienced in the general lesson content. There were 18 subjects in each category. Instructors were randomly assigned to one of three lesson conditions (i.e., the original version, the composite practice version, or the cognitive version). A pretest-posttest design was used.

As in the previous study, a performance test was designed that included most of the major lesson goals and procedures. The performance test was essentially a graphic simulation of a series of tactical situations composed of the scenarios presented during the lesson. It required the trainee to take the appropriate actions, based on knowledge of symbology and procedures. The performance test was made up of 18 procedures. Correct performance of each procedure was scored manually by an experimenter standing behind the subject. In addition, there were paper and pencil tests for recognition and recall of the lesson material.

Results and Discussion

The goal of the analysis was to determine the effect of lesson structure on procedural learning. The criterion for having learned a procedure is error-free performance of that procedure on the postlesson simulation test. The criterion for having learned all the procedures presented in the lesson is error-free performance on the entire simulation. Since time was limited to one pass through the lesson, it was not feasible to train each subject to asymptotic performance and then to measure the amount of training time required to reach criterion. In the ET environment, the trainee has a limited time available for training and should, therefore, get as much as possible out of each session. An alternative to repeating the lesson until some criterion is reached is to look at how close subjects can come to that criterion after one pass through the lesson. Of the 36 subjects, 27 made two or less errors (of 18 possible) on the post lesson performance test, the remaining nine made five or more errors. One measure of the relative effectiveness of the lesson structures is the percentage of subjects in each group who reached the given level of performance. Of the 27 subjects who made two or less errors, 11 were in the cognitive group (91%), nine in the composite practice group (75%), and seven in the original group (58%)

The three groups were compared for effectiveness of training as measured by performance error difference scores, that is the number of errors made on the pretest minus the number of errors made on the posttest. For all three groups, the decision to end the lesson was controlled by the subject. They always had the option of going back over any portion of the lesson before ending the lesson and attempting the performance test. An analysis of variance was performed on the difference scores for each subject in each group; the means are shown in Figure 3. Main effects were found for experience ($p<.01$) and for lesson structure ($p=.05$). There was no significant interaction.

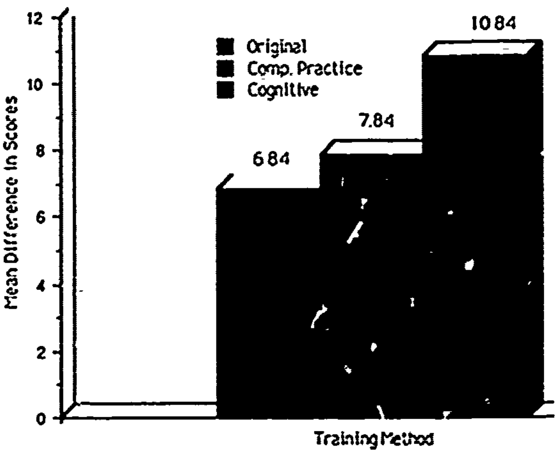


Figure 3. Mean reduction in performance errors from pretest to post test

The significant overall F allows us to reject the null hypothesis that there are no differences in performance that can be attributed to lesson structure. An analysis of differences in time to complete the lesson did not yield significant results and therefore do not account for differences in performance. In order to further assess the validity of our hypotheses, that both the declarative (composite practice) and the cognitive structuring of the exercises would improve performance over the original version, and to further identify the source of the variance, a pair of contrasts comparing the composite practice and the cognitive groups to the original group were computed. The differences between the group means were evaluated using Duncan's multiple comparison procedure. The difference between the means of the composite practice group and the original group did not fall within the significant range. The difference between the cognitive group means and original group means was found to be

significant at p<.05. Therefore, our hypothesis was confirmed regarding the improvement which cognitive restructuring produced relative to a standard instructional systems design approach to curriculum content. However, perhaps more important than how the declarative information is structured is the manner in which the opportunities to practice using that information and receive feedback are distributed throughout the lesson.

Having further isolated the source of the main effect to the variability between the cognitive and original versions of the lesson, the next logical question is to ask whether this difference is equally distributed over both levels of experience? To evaluate the hypothesis that the approach will be more pronounced for less experienced subjects, comparisons were computed between the cognitive and original cells for both the less experienced and the more experienced groups. The difference between cell means was found to be significant for the less experienced group (p<.025), but not for the more experienced group. Figure 4 provides a comparative illustration of performance error difference means for each of the experience level by lesson structure cells.

While the lesson structure had an impact on the ability to perform procedures correctly, there was little effect on declarative recognition or recall. An analysis of the variance between the three groups in recognition test difference scores, and in recall test difference scores yielded no significant effects. That is, the difference in lesson structure did not influence the trainees' ability or motivation to memorize information as much as it did their ability to perform procedures. This can probably be attributed to a general 'learning by doing' principle and to the opportunity to receive immediate feedback. When we present opportunities for interaction with a system during training, we are emphasizing the importance of the information presented in the interactive format, and de-emphasizing all other information. The student attends to just that information with which he is encouraged to practice, and for which he is provided feedback.

CONCLUSIONS

The results of these experiments support the argument that procedural knowledge and skills can be acquired more effectively when instruction is presented in accordance with cognitive learning principles interpreted within a knowledge compilation framework. Those instructors with NTDS experience came to the training session with device specific procedural knowledge as well as general domain declarative and procedural knowledge. They are more highly tuned to the training environment and can therefore easily apply existing knowledge and skills to organize and interpret the new instructional material within the training environment. Differences in organization of the instructional text do not significantly enhance their ability to encode and apply the relevant information. In addition, these instructors have existing console specific procedural knowledge which probably makes practice at the microstep level unnecessary.

Those instructors without NTDS experience came to the training session with general domain knowledge but without device specific procedural knowledge. In fact as illustrated by their comments (e.g., "I'm not used to doing it this way."), their methods for achieving specific task goals often were in conflict with the target methods of the training session. The opportunity to interpret and apply declarative information at each substep before going to the composed level enhanced the effectiveness of the training lesson for these individuals. In general, while the effect was much more dramatic with entry level trainees, even these well trained and motivated individuals learn the procedural methods more effectively when the opportunity is provided to perform and practice those methods, and receive feedback, at each individual step in the procedure.

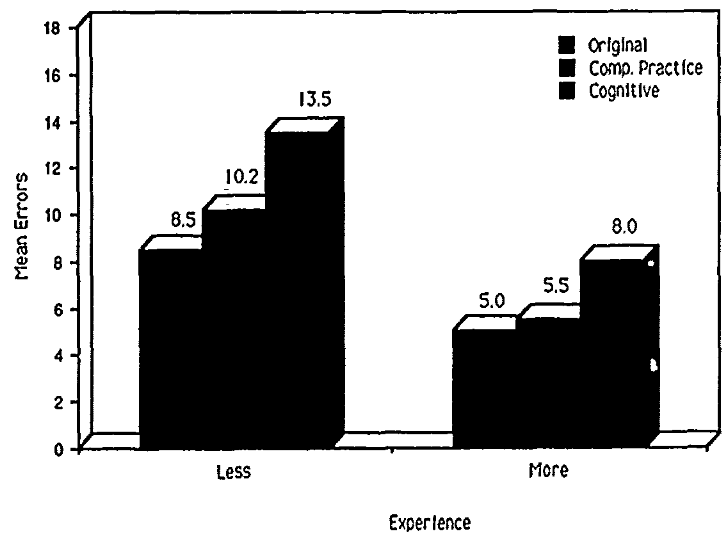


Figure 4. Mean pretest-post test performance difference for each of the experience levels by lesson structure cells

This approach can be applied to other embedded training systems and is currently being transitioned to the Aegis Computer Assisted Submode Training (CAST) system.

REFERENCES

Anderson, J. R. (1981). *Cognitive skills and their acquisition.* Hillsdale, New Jersey: Lawrence Erlbaum Associates.

Anderson, J. R. (1983). *The architecture of cognition.* Cambridge, MA: Harvard University Press.

Anderson, J. R. (1986). Knowledge compilation: The general learning mechanism. In R. S. Michalski, J. G. Carbonell, & T. M., Mitchell, (Eds.). *Machine learning: An artificial intelligence approach* (Vol. 2, pp. 289-310). Los Altos, CA: Morgan Kaufmann.

Anderson, J. R., & Jeffries, R. (1985). Novice LISP errors: Undetected losses of information from working memory. *Human-Computer Interaction,* 22 403-423.

Anderson, J. R., Conrad, F. G., & Corbett, A. T. (1989). Skill acquisition and the LISP tutor. *Cognitive Science,* 13, 467-505.

Anderson, J. R., Boyle, C. F., Corbett, A. T., & Lewis, M. W. (1990). Cognitive modeling and intelligent tutoring. *Artificial Intelligence,* 42, 7-49.

Bovair, S., & Kieras, D. (1990). Toward a model of acquiring procedures from text. In Bass, Kamil, Rosenthal, and Pearson (Eds.), *Handbook of reading research* (Vol. II). New York: Longman.

Carolan, T.F., Williams K.E., and Moskal P.D. (1991). Structuring Curriculum for Embedded Training: An Evaluation in an Operational Setting. Final technical report submitted to NTSC.

Carroll, J. M. (1990). *The Nurnberg Funnel: Designing minimalist instruction for practical computer skill.* Cambridge, MA: MIT Press.

Card, S. K., Moran, T. P., & Newell, A. (1983). *The psychology of human-computer interaction.* Hillsdale, New Jersey: Lawrence Erlbaum Assoc.

Chi, M. J. H., Bassok, M., Lewis, M. W., Reimann, P., & Glaser, R. (1989). Self explanations: How students study and use examples in learning to solve problems. *Cognitive Science,* 13 (2), 145-182.

Glaser, R., & Bassok, M. (1989) Learning theory and the study of instruction. *Annual Review of Psychology,* 40, 631-666

Hayes-Roth, F., Klahr, P., & Mostow, D. J. (1981). Advice taking and knowledge refinement: An iterative view of skill acquisition. In J. R. Anderson (Ed.), *Cognitive skills and their acquisition* (pp. 231-253). Hillsdale, NJ: Lawrence Erlbaum Associates.

Johnson, W. L. (1991) Review of J. Psotka, L. D. Massey and S. A. Mutter *Intelligent Tutoring Systems: Lessons Learned.* Hillsdale, New Jersey: Lawrence Erlbaum Associates.

Kieras, D. E. (1987a). *The role of cognitive simulation models in the development of advanced training and testing systems.* Office of Naval Research Technical Report No. 23 (TR-87/ONR-23).

Kieras, D. E. (1987b). *What model should be taught: Choosing instructional content for complex engineered systems.* Office of Naval Research Technical Report No. 24 (TR-87/ONR-24).

Kieras, D. E. (1988). Towards a practical GOMS model methodology for user interface design. In M. Helander (Ed.), *Handbook of human-computer interaction* (pp. 135-157). North Holland: Elsevier Science Publishers.

Larkin, J. H., & Simon, H. A. (1987). Why a diagram is (sometimes) worth ten thousand words. In M. D. Ringle, J.G. Greeno, and W. G. Lehnert (Eds.), *Cognitive Science* (Vol.11, pp. 65-100). Norwood, NJ: Ablex Publishing Corp.

Newell, A., & Simon, H. A. (1972). *Human problem solving.* Englewood Cliffs, NJ: Prentice Hall.

Reder, L. M., Charney, D. H. & Morgan, K. I. (1986). The role of elaborations in learning skill from an instructional text. *Memory and Cognition,* 14, 64-78.

Reif, F. (1989). Interpretation of scientific or mathematical concepts: Cognitive issues and instructional implications. *Cognitive Science,* 11, 395-416.

Rosenbloom, P. S., & Newell, A. (1986). The chunking of goal hierarchies: A generalized model of practice. In R. S. Michalski, J. G. Carbonell, & T. M. Mitchell (Eds.), *Machine learning: An artificial intelligence approach* (Vol. 2, pp. 247-288). Los Altos, CA: Morgan Kaufmann.

Williams, K. E., & Reynolds, R. E. (1989) *Embedded training instructional technology enhancement* (NTSC Tech. Rep. 89-034). Orlando, FL: Naval Training Systems.

Williams, K. E., Reynolds, R. E., & Carolan, T. F. (1989). *Knowledge engineering domain knowledge for an intelligent computer aided instruction system.* In the proceedings of the Second Florida Artificial Intelligence Research Symposium, April 1989.

Williams, K. E., Reynolds, R. E., & Carolan, T. F. (1990). *Automated Adaptive Instruction for Embedded Training.* In the proceedings from the Interservice/Industry Training Systems Conference (I/ITSC) 12th annual meeting, November 1990.

Williams, K. E., Reynolds, R. E., Carolan, T. F., Anglin, P. D., & Shrestha, L. B. (1989) *An evaluation of a methodology for cognitively structuring and adaptively sequencing exercise content for embedded training.* (NTSC Tech. Rep. 89-035), Orlando, Florida: Naval Training Systems Center.

SPEECH RECOGNITION IN REALTIME TRAINING: METHODS OF RECOGNITION RECOVERY

Lynne M. Pusanik

Logicon, Inc.
Tactical and Training Systems
San Diego, California

— ABSTRACT —

In the last twenty years, speech technology has progressed from very small vocabularies spoken with isolated speech to very large vocabularies using continuous speech. Since the field of speech technology is relatively new, the focus has been on achieving perfect recognition, the "expected" result. Unfortunately, misrecognition and inappropriate commands produce "unexpected" results. Identification and treatment of these results can be nearly as important during training as achieving the "expected" result. "Unexpected" results inherent in using speech recognition can be used to enhance training. This paper classifies different types of unexpected results and presents methods for dealing with them. Emphasis is placed on optimizing the training environment versus optimizing the operational environment. The rejection of recognition results with low confidence levels is discussed as well as its applications. Examples from Logicon's Tower Operator Training System (TOTS) and Advanced Shipboard ATC Training System/ Shore Based Radar ATC Training System (ASATS/SATS) are presented.

— BACKGROUND —

As more training systems incorporate speech recognition, the need to effectively address problems inherent in speech recognition increases. Thus, the challenge today is to identify these inherent problems and to design effective training systems despite them.

These problems may be expressed in a series of questions: what happens when students do not adhere to the trainer's phraseology? Or, what happens when students give commands which do not make sense? This paper will address these questions and propose some strategies, giving examples from Logicon's Tower Operator Training System (TOTS) and Advanced Shipboard ATC Training System / Shore Based Radar ATC Training System (ASATS/SATS). The TOTS program presents students with a realistic, wide-angle visual simulation of an airport scene for procedural training. All controls, displays and communication equipment found in a control tower cab are replicated in the TOTS environment The ASATS/SATS program consists of a series of air traffic control laboratories which model both shipboard and shore-based ATC radar environments. Both TOTS and ASATS/SATS use speech recognition and speech synthesis to allow students to communicate directly with aircraft. The examples presented from these programs are from a combination of integration during development, contractor and government testing both in-plant and on-site, and operational experience.

- CLASSIFICATION AND IDENTIFICATION -

Examining the results from a speech recognizer can help to answer the preceding questions. For the purpose of this paper a result will be defined as any action or response by the training system in reaction to a spoken command. An expected result will be defined as the result generated by the system in response to a syntactically correct command or in response to a logically correct command. Any other kind of result is an unexpected result. These unexpected results are the product of improper use of the training phraseology or misrecognitions due to stress, speed or vocal inconsistencies. Unexpected results can be classified in terms of the types of commands which generated them (remember the saying "Garbage in, garbage out"). Commands generating unexpected results can be classified into four different groups. inappropriate commands, contradictory commands, misspoken commands, and low confidence commands. A summary of these commands may be found in *Table I.*

Table 1. Summary of Command Classifications

Command Type	Characteristics	Examples
Inappropriate	Legal, but does not make sense or is physically impossible	"Expected approach time 20" when time now is 25
Contradictory	Subset of inappropriate; contradicts previous command	"Make approach straight in runway 28R", "Cleared to land runway 28L"
Misspoken	Syntactically incorrect or incomplete; includes verbal placeholders, coughing, sneezing...etc.	Any spoken command which is not found in the legal training phraseology
Low Confidence	Result returned by the speech device with a low confidence score	Dependent on confidence threshold

Inappropriate Commands

Inappropriate commands are defined as commands which are syntactically legal, but which do not make sense during a training exercise. One example of an inappropriate command is to tell one aircraft to tank with a second aircraft, when neither aircraft is a tanker. Commands such as this can result in seemingly strange actions by the trainer if results from these commands are enacted as they were spoken, therefore, it is desirable for the trainer to identify these commands so that other, more appropriate actions can ensue.

These commands can be identified by incorporating specific knowledge about the training exercise into the training system. By giving the trainer more knowledge about certain events, the trainer can identify commands which do not follow the prescribed pattern. In other words, the trainer needs the ability to determine the "appropriateness" of a command before processing. Some examples of relevant knowledge for an air traffic control trainer would be information defining marshal and holding patterns, and information regarding tanking procedures. Verbal commands can be used to supplement this information with marshalling points, holding points, and identification of tankees and tankers. With a knowledge of valid marshalling and holding patterns the system can identify any commands using invalid marshalling points, holding points, or tankees and tankers. By identifying invalid inputs, the training system is able to identify inappropriate commands.

Identifying physical incongruities also contributes to identifying inappropriate commands. For instance, in Logicon's SATS, the student may tell an aircraft an expected approach time. If this occurs too early, then the aircraft would physically be incapable of completing the command as directed (barring teleportation). By utilizing information regarding the physical characteristics of the target and the environment, the training system is able to identify inappropriate commands.

Contradictory Commands

Contradictory commands consist of a specialized subset of commands within inappropriate commands. Commands which contradict information given in a previous command are defined as contradictory commands. Some contradictory commands occur legitimately within a training exercise. For instance, simply because an aircraft was previously told to turn right is no reason not to tell the aircraft to now turn left. Other commands, however, may change information, which the trainer should question. An example of this, from Logicon's TOTS, would be an initial transmission of "Make straight in runway 28 right," followed some time later by "Cleared to land runway 28 left." In this case, the student told the aircraft to make its approach to runway 28 right, but later cleared the aircraft to land at runway 28 left. The aircraft may not physically be in a position to comply with the student's commands; the student's second command should be questioned.

These commands are identified in the same way as inappropriate commands, by carefully defining all possible events within a training session. Some of this information is provided dynamically by the students' commands. For example, the runway is identified for the approach. Discrepancies from the trainer's updated knowledge of approaches may now be identified. For instance, using the earlier example, if the student originally defines an approach to one runway, and in a later transmission clears the aircraft to another runway, the training system can identify the command as being contradictory.

Misspoken Commands

Unfortunately, when students make mistakes in speaking to a trainer, they do not always make mistakes which are part of the "legal" training vocabulary. Misspoken commands

are commands made by the student which are either not syntactically correct, or which are incomplete. These types of commands include verbal placeholders (drawing out a word while trying to think of what to say) and coughs, sneezes, etc. The most important feature of these types of commands is that it is not possible to predict, with any certainty, the recognition which will result. For students first using a trainer, this type of command is very common. During government testing on various air traffic control trainers, these types of commands were the most common mistakes made by experienced controllers. Generally, the training phraseology is more restrictive than the operational phraseology, even experienced personnel will misspeak some commands initially.

Incomplete, or partial, commands can easily be identified by the training system, since the information to process a partial command is incomplete. For example, in both ASATS/SATS and TOTS, the command "Turn left heading 015" is part of the legal phraseology. In order for the turn to be processed by the trainer, the entire command must be spoken. If a student said only "Turn left heading 1" and then ended the transmission, the trainer knows that an incomplete command has been made because the heading must consist of three digits. Government preliminary inspection (GPI) of the Radar Air Traffic Control Facility (RATCF) of SATS indicates that 12% of all misrecognitions are due to incomplete commands. This includes both commands which were spoken incompletely and improper use of the push-to-talk (ptt) which resulted in commands being recognized incompletely.

With the exception of incomplete commands, misspoken commands can only be identified by monitoring students' transmissions. This can be facilitated by installing a recording system in the trainer which allows for review of student transmissions at a later time. Comparing the spoken commands with the legal training phraseology will yield any misspoken commands. Often, trends in misspeaking can be identified for both a single speaker and for a group of speakers. For example, in the SATS phraseologies the phrase "Expected approach time 4 5" is a legal command. Also, several commands which begin with "Expect" are legal. Reviewing tapes of experienced controllers using the trainer have shown that a particular controller says "Expect approach time 4 5" in place of the legal command. As a result, this command is often misrecognized. A common mistake found among a large group of controllers using ASATS/SATS is the use of the word "disregard." "Disregard" is a command used often in the operational environment, however, its use is not permitted in the ASATS/SATS environments.

Reviewing tapes from RATCF's GPI shows that 30% of all misrecognitions are due to misspoken commands. These misspoken commands were found to be commands delivered in realtime in a manner inconsistent with offline training, commands using invalid phraseology as defined for the training environment, misuse of the word "correction" as defined for the training environment, and incomplete commands.

Unfortunately, these types of commands can only be identified manually. A speech recognition device cannot indicate if the recognized result is actually the same as the spoken command because it does not know. The speech recognition device provides only the closest match to what was spoken.

Low Confidence Commands

Low confidence commands are those with results deemed questionable by the speech recognition device. These may be commands which are syntactically correct, but the speaker stumbled over the words. These commands may not be a part of the legal phraseology (i.e., misspoken commands). These may be legal commands, where the speech recognition device has greater confidence in some of the words spoken in a command, and less confidence in others. A command which returns a result with a low confidence value is not necessarily incorrectly recognized. It simply has a greater probability of the result not matching the spoken command. However, the result which was returned, regardless of its confidence level, is still the most probable result possible according to the speech recognition device.

Not all speech recognition devices have the capability of measuring confidence levels in recognition results. And, of the devices which do have this capability, the amount of control the developer has over the rejection or acceptance of low confidence commands varies. Some devices use internally defined parameters to indicate the rejection of any recognized results below a defined confidence level. Other systems allow the developer to define the acceptable confidence level, and whether or not results should be rejected. With most speech recognition devices, if rejection is enabled, then the device will not provide any results for a command whose confidence is below the acceptable level.

Other speech recognition devices allow the developer to disable rejection of low confidence commands, yet the device still provides rejection and acceptance scores the developer can use. The developer can, with significant research, develop algorithms for using these scores with particular applications. For instance, once a threshold confidence level has been arrived at, the developer must determine the thresholds to reject an entire command. The developer needs to determine how many words need to be rejected in a command before the entire command is rejected and whether all words in a command are equally significant. The algorithm may vary depending on the number of words in a command or on the action of the command. For instance, if a command is recognized as an advisory command (i.e., one with no action associated with it) there is not much point in analyzing the command for rejection.

— METHODOLOGIES —

Once a result has been identified as a particular type, one can determine methods for treating the result. While the result from the speech recognition device may not change, the way in which the training system responds to the result my be modified to enhance training. The methodologies presented are summarized in *Table 2*.

While acting upon the understood command teaches the student some lessons (namely, be careful of what is said), in most cases training will be improved if mistakes are pointed out to the student. When dealing with any future timed event (for example, reports), implementing contradictory commands can be very confusing. For these reasons, training is more likely to be enhanced by flagging them in some way rather than by implementing these commands.

Table 2. Summary of Methodologies

Methodology	Characteristics	Applicable Command Types
Non-Intervention	No special action	Inappropriate, Contradictory, Misspoken, Low Confidence
Phraseology Restriction	Restrict legal phraseology to appropriate commands	Inappropriate
Notification	Flag command at IOS or student position - use voice feedback for enhancement	Misspoken ("Say Again"), Contradictory ("Say Again Runway"), Inappropriate ("Unable"), Low Confidence
Instruction	Instructor points out mistakes to students	Misspoken
Rejection	Reject commands with scores below confidence threshold	Low Confidence

Non-Intervention

One option is for the trainer to act on the command as it was understood. This type of action can be a two-edged sword. The student will learn that the trainer will do exactly what it is told to do, not what the student meant it to do. In this way, this option can benefit training. Many inappropriate and contradictory commands can be successfully treated in this way. For example, if a student broadcasts "Turn right heading 120, Climb and maintain flight level 100, turn left heading 130," the student may intend the aircraft to complete the first turn, change altitude, and then complete the second turn. However, the trainer will begin the first turn, and start the altitude change as soon as it is understood (while still turning), and the second turn will override the first turn causing the aircraft to stop the first turn and perform the second turn. Although the trainer performed as the student directed, the performance was not the student's intention. The developer should keep in mind though, that the student may not be able to identify why the command was incorrect. By performing the action of a poor command, the trainer may get the student into a situation from which it is difficult to recover. Selection of this option, therefore, should be weighed against the expertise of the typical student.

With low confidence commands, non-intervention is not as poor as it sounds. Since a 'low confidence' result is simply a result with a lower probability of being correctly recognized, the probability still exists that this result was correctly recognized. And, as previously mentioned, this result is still the best result the speech recognition device can generate.

Phraseology Restriction

Another option for the developer is to try to eliminate the possibility of verbal mistakes. This option is particularly effective for resolving inappropriate commands. Some inappropriate commands can be eliminated completely by restricting the legal phraseology. When the user and developer define the phraseology, effort should be made to include only vocabulary which is pertinent to training. For example, ASATS phraseology consists of three phraseologies which are specifically designed for different air traffic control positions. There is a vocabulary for the approach position, for the arrival position and the final position. As a result, commands which are specific to the arrival position are not included in the final position's vocabulary. By predefining the legal phraseology, the developer has already eliminated a large number of commands which are inappropriate to the training system.

In the same way, inappropriate commands can be restricted from specific events which occur at each position. For example, in RATCF, an aircraft on a final precision approach does not expect a change in heading of more than 30 degrees. By limiting the amount of incremental turns allowed in the vocabulary to 30 degrees, the student will not be able to generate a result for a turn of more than 30 degrees. Even if a student commands the aircraft to turn 50 degrees, the speech recognition device will interpret the incremental change to be thirty degrees or less, as defined in the training phraseology. In the event that the student does need to turn the aircraft further than 30 degrees, the student is allowed to give explicit heading changes; for example, "Fly heading 120." Thus, the student is not restricted in controlling the aircraft, but is restricted only in the choice of commands used to control the aircraft.

Notification

If a command is appropriate in some cases, but inappropriate in others, eliminating the command from the phraseology is not a reasonable solution. Another method for handling inappropriate commands is to flag the command in some way. Flagging a command indicates a potential problem to either the student or the instructor. Commands can be flagged at the Instructor Operator Station (IOS), or at the student position. Voice feedback can be used to enhance the notification of commands at either position.

Flagging commands at the IOS alerts the instructor to any resulting difficulties the student may experience. If the instructor is in a position to intervene, situations which the student cannot yet handle can be averted. Feedback directly to the student allows the student to immediately correct a mistake. With the use of voice synthesis, feedback to students can be as realistic as the operational environment.

TOTS and ASATS/SATS use a combination of these methods. Speech recognition feedback is available at the IOS. This feedback prints the recognition results for a student on a screen at the IOS. Besides providing information on speech recognition, it also identifies the student speaking to an inactive target or broadcasting on an incorrect frequency.

Information regarding the target's position, heading, speed and other vital statistics is also accessible at the IOS. As a target performs a turn, the instructor can watch the heading change at the IOS In RATCF, if a student clears an aircraft for landing, the landing intentions displayed at the IOS indicate a full stop. If the student later clears the air craft for a low approach, the landing intentions displayed at the IOS change to indicate a low approach. In this way the instructor can monitor typical contradictory commands made by the student.

In ASATS/SATS and TOTS, students are notified of inappropriate commands by pilot responses of "Unable." If the student commands an aircraft to do something the aircraft is physically incapable of doing, the aircraft's pilot will respond with "Unable." For example, if a target is making a no-gyro final approach and the student transmits "Turn right heading 110," the pilot will respond with "Unable" (since a no-gyro approach prohibits turns to hard headings). In other words, the trainer prevents actions from occurring for an inappropriate command and identifies the command to the student.

The TOTS trainer also utilizes vocal responses for handling specific contradictory commands. Using an earlier example, if a student transmits "Make approach straight in runway 28 right," and then later transmits "Cleared to land runway 28 left," a pilot voice will respond with "Say again runway." This response informs the student that the target was expecting a clearance for a different runway. If the student repeats "Cleared to land runway 28 left," then the target will change its approach to runway 28 left. If, however, the student mistakenly gave an incorrect runway in the clearance command, the student can recover from the mistake without any adverse reactions from the trainer. This feature allows the trainer to identify a possible error to the student, and also allows the student to correct the error while maintaining realism in the trainer.

While not all misspoken commands can be identified by the trainer, incomplete commands can be identified. In ASATS/SATS and TOTS, these commands are resolved by generating a pilot response of "Say again." This informs the student that a command was not completely recognized; now, the student can repeat the command correctly.

Rejection of low confidence commands may be combined with notification by using vocal responses. This response could be either a pilot response or as a "system" voice. Ideally, a "system" voice would be a voice other than the expected operational voices. In an air traffic control trainer, a possible pilot response might be "Say again." A "system" voice could synthesize a message to the student indicating that the last command was rejected.

Instruction

Once instructors or developers have identified trends in misspeaking, the students should be informed of the tendencies. Ideally, students should be corrected as soon as misspoken commands are identified to prevent the habitual use of improper phraseology. Since the training system cannot reliably identify most misspoken commands, the training system cannot directly intervene to correct misspoken commands. Some speech recognition devices, however, may provide additional information regarding the confidence level of the recognized results. A method for managing these results is treated as a separate case.

Rejection

The treatment of low confidence commands depends on two items: the ability of the speech device to identify low confidence commands, and amount of control which the developer has in rejecting low confidence commands. Low confidence commands may either be rejected automatically, or rejected in combination with notification.

The first option is to simply reject the result from a low confidence command with no feedback. Some speech recognition devices operate in this manner as a matter of course. This rejection could confuse the student since the system would neither react to the command, nor would it indicate any reason for the lack of action. A better solution would be to reject the command while also providing notification of rejection.

Low confidence commands are currently not being rejected in any of the TOTS, or ASATS/SATS trainers. This is primarily because a proven method for using rejection scores was not available at implementation. In addition, analysis for rejection can be time-consuming in a realtime environment. Potentially, in an environment where many commands are spoken in a small time period (such as the Final position on ASATS/SATS), the time taken to reject low confidence commands and the subsequent pilot response could impact other commands. Also, other methods of resolving unexpected results have proved more than sufficient for enhancing training capability. While the idea of rejecting low confidence results is viable, the methods for rejecting low confidence results are still under development.

— CONCLUSION —

ASATS/SATS and TOTS have successfully integrated speech recognition into Air Traffic Control simulators and use many of the methods discussed for identifying and resolving unexpected results. The success of ASATS/SATS and TOTS has resulted from incorporating many of the strategies discussed including non-intervention, phraseology restriction, notification and instruction. Using these methods during initial government testing of RATCF allowed specific types of misrecognitions to be quantified; 7% of misrecognitions were identified as inappropriate or contradictory commands, and another 30% of misrecognitions were identified as misspoken commands. The ability to quantify these misrecognitions has led to the identification of areas in which a student is having problems.

By identifying unexpected results and methods to govern them, a training system can be more than just a simulator. The training system has the potential to teach the student by identifying, but not necessarily correcting, mistakes which the student makes.

ABOUT THE AUTHOR

Lynne Pusanik works for Logicon, specializing in speech technology. Currently she is involved with speech recognition and voice enrollment/training for the Advanced Shipboard ATC Training Systems, Shore Based Radar ATC Training Systems and for the Tower Operator Training System. She graduated from Texas A&M University with a master's degree in bioengineering and a bachelor's degree in mechanical engineering.

VIRTUAL REALITY: THEORETICAL AND PRACTICAL IMPLICATIONS

Richard A. Thurman
Joseph S. Mattoon
Armstrong Laboratory
Aircrew Training Research Division (AL/HRAU)
Williams AFB, AZ

ABSTRACT

The concept of virtual reality and the wave of research and development accompaning it are creating new forms of simulation that may lead to fundamental improvements in simulation-based training. However, because virtual reality is a relatively new concept within the training community, there seems to be a few misconceptions concerning what virtual realities are, how they are created, and how they can be used. In an effort to clarify readers' understanding concerning virtual realities, this paper examines three dimensions that will describe virtual reality's role and effectiveness in simulation-based training.

INTRODUCTION

Virtual reality refers to both the experience of residing in an artificial environment and the medium that makes such an experience possible. As an experience virtual reality offers a strong sense of "presence" (Zeltzer, 1990) in a synthetic environment, unlike a movie which may portray an alternate reality but cannot give viewers the compelling sense that they are actually present within it. As a medium, virtual reality is a type of interactive computer-based simulation controlled in part by the user (Walsher, 1991). In a virtual reality, the user is included as part of the simulation. This represents a break from traditional computer-based games and simulations which provide interactive components but do not include the user as part of the simulation.

Virtual reality is a loosely defined concept, represented so far by a number of rather simple prototypes (Helsel & Roth, 1991). Yet, the concept holds promise for applications in military training and other types of computer-assisted instruction. Virtual reality research is leading to new insights in modeling, simulation, and multisensory human computer communication (Burg et al., 1991). However, there are indications that virtual reality technology is often misinterpreted as a single technological innovation associated with helmet mounted displays, or sometimes with input devices such as DataGloves. The technology is also mistaken for particular hardware and software implementations in much the same way that particular computer processors (e.g., lisp machines) and languages (e.g., Prolog) were mistaken for artificial intelligence technology. In an effort to clarify readers' understanding of virtual reality's potential for training applications, this paper describes three components that will determine its role and effectiveness in simulation-based training. The theoretical and practical implications of "verity," "degree of integration," and "natural vs. artificial interface" are defined below.

VERITY

The verity dimension of a virtual reality refers to the degree that our natural environment is simulated. The word verity is derived from a Latin term, "veritis," which means "true to life." It is used here to denote a continuum of simulation experiences that range from simulating, as much as is humanly possible, the physical world as we know it to simulating a totally invented world.

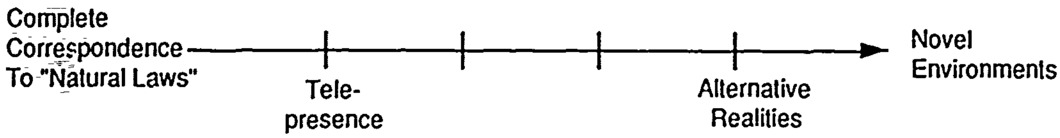


Figure 1. The Verity Scale

Figure 1 depicts the verity of virtual realities ranging from "telepresence," the representation of the real world in real time, to "alternative realities," which are completely novel environments . An example of telepresence is depicted by remote-controlled, undersea robots that provide a view and a way to manipulate objects in sea environments (Spring, 1991).

NASA's Ames Research Center, has developed a helmet mounted display which, when worn by astronauts inside a space station, enables them to see through a robot's "eyes" outside the station. As an astronaut's head turns, the robot's camera eyes simultaneously swing in the same direction (Foley, 1989). Astronauts can make repairs without leaving the safety of the space station. Telepresence literally allows users to be in more than one place at the same time.

Further toward the novel environments end of the continuum are simulations that go beyond reality. These virtual realities model, in a tangible form, abstract information such as mathematical equations, vortices, shock systems, and flow patterns. In their visual form such information is commonly represented by lists of or matrices of numbers, but when converted into graphic images, the information undergoes a qualitative change that brings the human visual system into play. Our ability to recognize, categorize, and analyze visual and aural patterns goes far beyond our ability to deal with purely numeric data (DeFanti, Brown & McCormick; 1989). This form of virtual reality may empower human users to overcome problems of scale in studying and manipulating objects such as atoms, molecules, DNA, brain maps, or entire galaxies (Foley, 1989).

Finally, at the far end of the verity scale lie those simulations which have little basis in physical reality. These virtual realities are represented by environments that have no physical counterparts and are governed by laws originated by the developers. In such virtual realities, the laws that govern the universe as we know it may be modified, suspended, or contradicted (Spring, 1991). For instance, mathematicians create computer-based models of curved, or hyperbolic, space, where the rules of geometry differ from those we normally encounter. As an example, in hyperbolic space the sum of the angles within a triangle is less than 180° (whereas the sum is exactly 180° in ordinary Euclidian space). Despite the fact that these "realities" exist only in the minds of physicists and mathematicians, they are useful in comprehending how objects act in curved space. Guided by carefully crafted instructions, computers can simulate familiar objects, display abstract concepts, and create new worlds beyond the confines of human experience - occasionally with surprising results. (Peterson, 1990)

INTEGRATION

Virtual realities are a type of interactive simulation which includes the human user as a necessary component. Virtual realities are fundamentally different from other interactive simulations in the way that the user is integrated within the computer simulation.

Viewed historically, there are three broad eras of human-computer integration. In early systems, the interface consisted of batch processing where the computer and user acted as completely separate entities. The user's job was to set up mathematical tasks to be performed in a linear sequence, code the information in a format that was interpretable by the computer (i.e., the computer program), start the program, and wait for the output. In this type of interface the user and the computer had very distinct and different roles. Those roles called for little, if any, human-computer integration.

The second era of human-computer interface came about when computers were developed that were capable of carrying on a dialogue with the user. For example, context sensitive help systems enabled the computer to assess users' error patterns and provide correct information to help them complete unfamiliar tasks. In the training arena computer-assisted instruction and computer-based training simulations were developed that made computers capable of assessing students' needs and providing feedback and other instructional assistance based upon student performance.

In the dawning era of human-computer interface, simulation technology makes it possible to go beyond simple assessment of user behavior. In such a system the dialogue is replaced by simultaneous interaction of the user and the computer within the simulation. Such interface designs can be typified by an interactive simulation where user input is simply one data source. In such systems, the simulation is less of a sequence of preprogrammed events. Rather, the simulation is created by constructing a computer world of animate and inanimate objects, defining the world's laws, and populating the world with consistently behaving characters. According to Burg et al. (1991), the idea is to create not just one sequence of events that replays itself every time the simulation is turned on, but to create a virtual world of objects which act and interact in an independent but emergent manner. In such a simulation, the computer defines what an object is and then insists that it remain true to its definition as it reacts to the evolving situation. Burg et al. states:

> The point here is that the programmer need only describe (declaratively) the objects in his world, without prescribing (procedurally) what each object will do at every turn. Written into the ... system are the laws which govern the virtual world, and behavior emerges naturally from object descriptions and the [virtual] world's natural laws. (p.5)

It is at this level of human-computer interaction where virtual reality is directed. Figure 2 graphically depicts the integration dimension of virtual reality.

Walsher's (1991) depiction of cybernetic simulation represents an appropriate description of an integrative simulation. According to Walsher, a cybernetic simulation is a dynamic model of a world filled with virtual objects that behave the same as their physical counterparts. In addition, certain objects (Walsher calls them "puppets") are controlled by the actions of human users (termed "patrons"). The patron's movements are monitored by sensors installed in DataGloves or DataSuits. When the patron moves, the puppet moves in direct relation. According to Walsher, the principle function of the cybernetic simulation, besides simulating a virtual world, is to maintain a tight feedback loop between the human user and the puppet. This will give the user "...the illusion of being literally embodied by the puppet (i.e., the puppet gives the patrons a virtual body, and the patrons give the puppet a personality)." (Walsher, 1991; p.35)

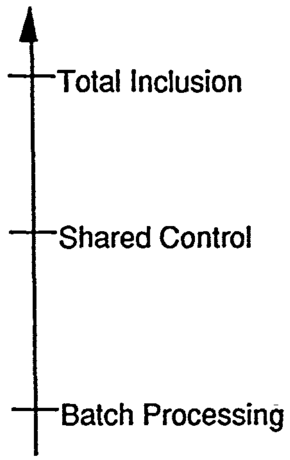


Figure 2. Integration Dimension of Virtual Reality

Figure 3 depicts the relationship between the patron and the puppet. The puppet monitors the patron through such sensors as data gloves, data suits, joystrings, or head trackers and acts on the patron through various effectors such as video displays, sound generators, resistance controllers, motion platforms, force feedback devices, and so on.

What makes Walsher's explanation of cyberspace useful is the way one thinks about sensors and effectors. Most interactive simulations are designed extrinsically, that is, from the user's point of view. Typically users conceive of themselves as standing outside the system and using "input devices" to put information in and "output devices" to get informa-

tion out. A cybernetic simulation, on the other hand, is defined intrinsically, that is from the puppet's point of view. Under an intrinsic point of view, the sensor in Figure 3 refers to a device through which a puppet acquires knowledge of events in the physical world. The same device would have been formerly called an input device. A puppet's effectors, on the other hand, are output devices in the old way of speaking. Walsher states:

> Generally, a patron affects virtual space through a puppet's sensors and learns of events in virtual space through a puppet's effectors. That is, a puppet's sensors are a patrons effectors, and a puppet's effectors are a patron's sensors. This can be confusing until you shift your thinking to the intrinsic viewpoint, and realize that discussion is always centered on a point of view from within the virtual space (the term patron, as another example, is used to suggest an actual visit to a place, such as a museum).

The ultimate form of this kind of human-computer integration will not require the use of a puppet. Instead the patron will be integrated directly into the simulation. A fully integrative system is depicted by the "Holodeck" simulations from the television series Star Trek: The Next Generation. The "Holodeck" views each patron as only one data source among many within the simulation (Spring, 1991). In addition, the patron needs no artificial sensors (e.g. DataGloves) to interact with the simulation.

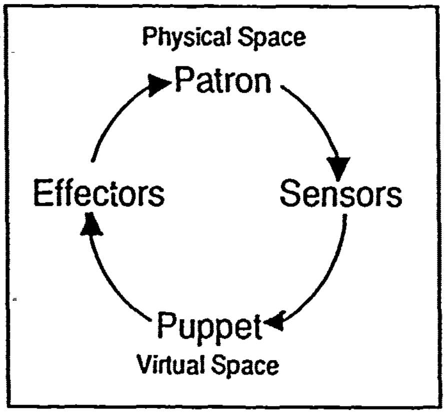


Figure 3. The Cybernetic feedback loop (adapted from Walsher, 1991).

NATURAL VS. ARTIFICIAL INTERFACE

The third dimension we have used to characterize virtual reality is the form of interface that enables the user and computer to interact within the simulation. On the low end of this continuum are artificial input/output devices such as a keyboard, mouse, or trackball. Moving up the continuum are body-mounted sensor/effector systems such as helmet mounted displays and datasuits. These systems represent a more direct and natural way of interfacing with the simulation. On the high end of the continuum the virtual reality system is capable of monitoring users' behavior, including voice commands and body position, without the use of body-mounted external input devices. Thus the computer and user interact quite naturally.

A good example of the interface dimension can be found in the field of oceanographic research. Because oceanographic researchers are dealing with an environment that, at times, can be quite inhospitable to humans, they have invented many devices to interface with our oceans and seas. These interface devices, like the human-computer interface, form a continuum ranging from the very artificial to the quite natural. At the artificial end of the continuum lie remotely piloted vehicles. These vehicles allow a researcher to "see" the underwater environment from the relative comfort of a computer terminal. The researcher need never get his feet wet! At the other end of the continuum, the natural end, researchers can completely immerse themselves in the medium by scuba diving. Somewhere in the middle of the continuum lies the use of underwater submarines which have remotely controlled robotic arms.

Likewise, on the low end (the artificial end) of the interface scale, a virtual reality may use some form of an external input device to control a "puppet" in a virtual space or, on the high end of the scale, the user may participate directly.

THE CUBE

Figure 4 shows that the three dimensions - verity, integration, and interface - when combined, form a three dimensional coordinate system which can be used to classify virtual reality systems*. All the

* The idea of depicting virtual reality in cube form and the classification system used here was inspired by Spring's (1991) and Zeltzer's (1991) articles.

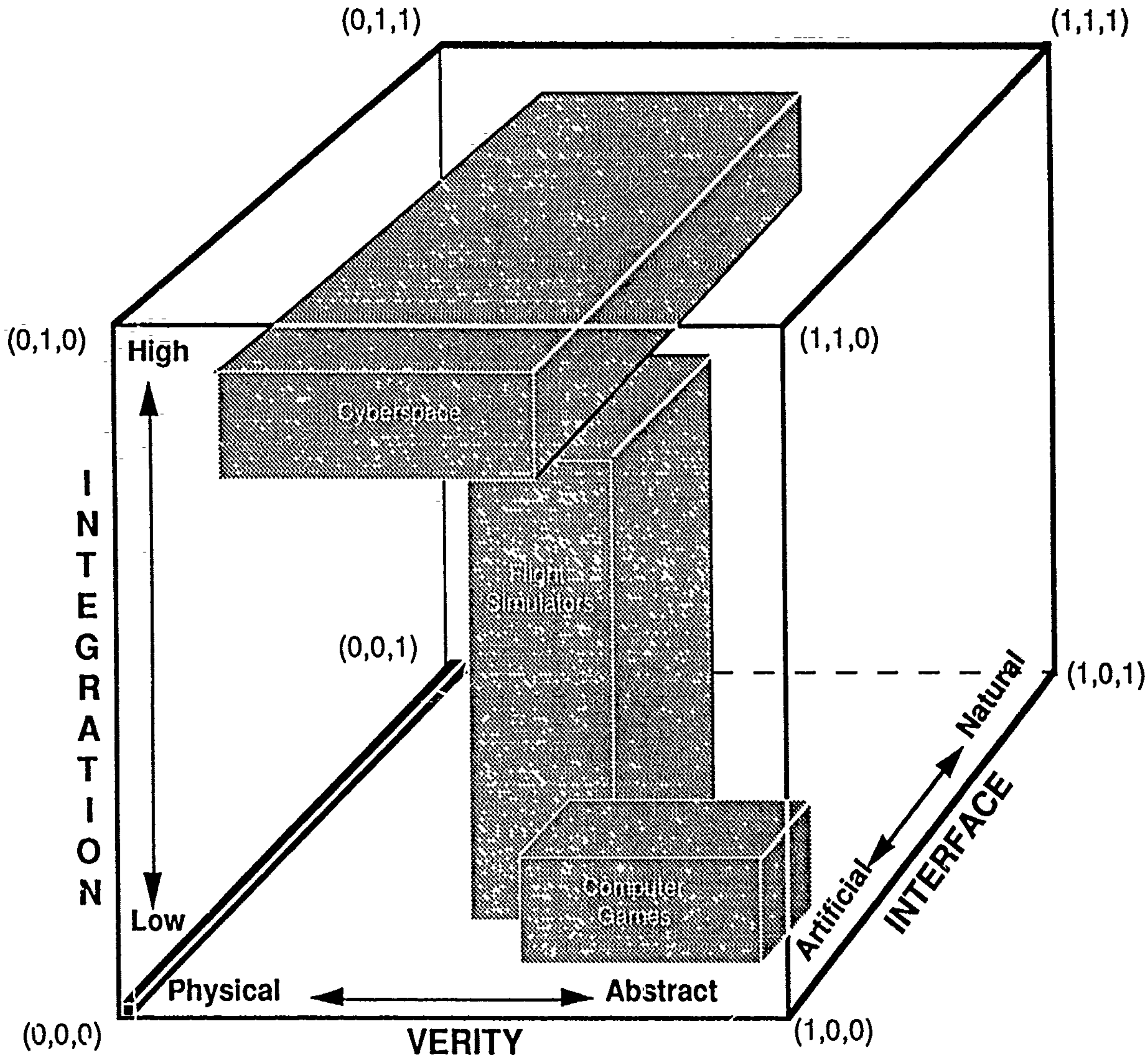


Figure 4. Dimensions of Virtual Reality

corners of the cube in Figure 4 have three numbers. Each number corresponds to the verity, integration and interface dimensions, respectively. For ease of interpretation the range for each dimension is scaled from zero to one. This lends itself to quantifying a virtual reality's salient characteristics. For example, one could describe a particular simulation by assigning a value of .33 for verity, .89 for integration and .65 for interaction.

At the origin (0,0,0) simulations are strictly numerical exercises, probably done in batch mode, where the user does not participate (except to read the printout when it is done). While computational simulations are very valuable to business and/or research, they are hardly virtual realities.

Position (1,0,0) is not much better from a virtual reality standpoint. This position exhibits the same limitations discussed above with the exception that this represents a situation where physical laws, as we know them have been sufficiently changed so as to represent an entirely new reality. For example, physicists sometimes write simulations which attempt to portray worlds which exist in five or more dimensions. Such work is very valuable to researchers who are trying to understand hyperbolic space (Peterson, 1990). Often they are able to print out graphic depictions of these artificial realities which

are really quite stunning in their visual and emotional impact. However, these simulations do not constitute a virtual environment. Actually, these simulations are straightforward in terms of input and output, require a minimum of interface with the simulation, and are little more than "batch mode" computations.

In contrast, positions (1,1,1) and (0,1,1) represent full implementations of virtual reality. At position (1,1,1) we have interactive simulations which are based on completely new laws of physics. In addition, users are completely immersed in such simulations, and have become another data point in the artificial world. They are completely free from any artificial means of experiencing the reality (e. g., DataGloves or helmet mounted displays). Position (0,1,1) represents simulations which are highly integrated and at the same time have a natural interface. In addition, simulations at position (0,1,1) are capable of fully simulating the laws of physics. At position (1,1,1) users could take a "pretend trip" to an imaginary moon, while at position (0,1,1) users would be able to walk on a fairly convincing representation of our own moon. Is such a system feasible? Is it doable? Probably not in the near future, but this concept brings home an important idea - if our virtual reality systems have enough data about the environment we wish to simulate, have enough data about the participant, and can process this data quickly enough, we can create new realities.

At position (0,1,0) simulations are accurate representations of physical reality where users are fully integrated with the simulation through artificial means. This position represents the telerobotic aspect of virtual reality - the command and control of remote physical devices by simulating a robot's view. Virtual robots would let users work in environments ranging from giant automated construction equipment to microsurgery. Users would operate in a virtual environment scaled to their real world, and the system would translate the user's actions to the scale of the application's real world via the computer's virtual world. The telerobotics aspect of virtual reality may very well have its greatest impact on situations where humans will need to operate safely and effectively in hazardous environments such as the ocean or outer space (Fisher, 1991).

At position (1,1,0) we find an interesting combination where simulations are based on alternate sets of physical laws but exhibit a high degree of integration through artificial interface devices. This position represents, among other things, the entertainment/gaming aspect of virtual reality. For example, computer games often place participants in a world of strange creatures where alternate laws of physics prevail. Virtual reality systems will expand the gaming concept to include the participant's entire body, rather than simply providing a joystick or keyboard as today's computer games do.

Finally, positions (0,0,1) and (1,0,1) represent situations that are familiar to all of us. At (0,0,1) we have simulations which are based in reality, with natural interfaces and with very little integration with the system. This point represents classic social and business simulations where participants play particular roles. This position also represents live theater. Simulations at position (1,0,1) have the added feature of being based in an alternate reality. Perhaps the best way to represent this position is by conceiving a precomputed conventional animation sequence viewed on a large screen where the user just sits back and enjoys the show. Currently there are a few amusement park rides which fit the description found in this corner. 'Tour of the Universe' in Toronto and 'Star Tours' at Disneyland were among the first entertainment applications of simulation technology and virtual display environments where approximately 40 people sit in a room on top of a motion platform that moves synchronously with a computer-generated display to simulate a ride through another universe (Fisher, 1991).

THE RELATIONSHIP BETWEEN VIRTUAL REALITY AND OTHER "SIMULATIONS"

The cube with its three dimensions - verity, integration, and interface - can be used as a tool for structuring our conceptualization of other kinds of simulations found in the entertainment and training arenas. Take for example the area of computer-based entertainment called "games and simulations." Figure 4 shows that most of these products fall into the lower right quadrant of the cube. These "simulations" are usually found on desk-top computers, dedicated entertainment systems, or in some cases, low-end work stations. Because of the lack of computing power of these hareware configurations, most of the simulations are fairly anemic in the verity dimension. They simply cannot model to any degree

of reasonable fidelity, the actual items they are attempting to simulate. In addition, computer-based games often deliberately try not to imitate "reality." After all fantasy is an important part of the appeal of computer-based games (Malone, 1981). In terms of interface, these "simulations" are usually limited to keyborad, joystick, or mouse inputs. The output is usually limited to a single CRT. In almost all cases, these are extremely artificial interface devices. In terms of inclusion, computer-based games and simulations have been almost always programmed by conventional methods. Users are no more a part of the simulation than if they were using a spreadsheet program.

Figure 4 also shows where flight simulators may fit into the cube. Notice that the space representing flight simulations takes up more volume within the cube. These simulations range from pc-based simulations on up to high-fidelity, full-mission simulators. As flight simulators become more sophisticated, in terms of programming techniques, attention to functional, physical, and psychological fidelity, and use of innovative interface techniques, they begin to take on more of the characteristics of virtual realities.

The cyberspace dimensions of virtual reality are depicted in Figure 4 as well. As discussed above, cybernetic simulations are designed from the simulated object's point of view (Walsher, 1991), may or may not require external (artificial) input and output devices, and model to a greater or lesser degree - depending on the application - real world objects.

As can be seen, the cube in Figure 4 allows us to conceptualize, and to some extent, quantify a simulation's qualifications as a virtual reality. The simulations represented in the cube are by no means exhaustive, the reader is invited to fill in the rest of the 'empty' space.

VIRTUAL REALITY AND SIMULATION RESEARCH

The cube, with its three dimensions, can also be used as a conceptual tool for structuring our understanding of virtual reality research. The verity dimension includes research and development of scientific visualization tools (DeFanti, Brown, & McCormick, 1989; Peterson, 1990), simulation fidelity (Alessi,1988), physics based modeling of objects (rigid and nonrigid), including anthropomorphic and jointed figure motion (Lee, Wei, Zhao & Bradler,1990; Wilhems, Moore & Skinner,1988; Carrington, Hughes, Burg & Xin,1991), and reactive planning (Gonzalez, 1991; Le, 1991). In terms of integration, ongoing research includes, among other things, object oriented, as well as constraint-based programming (Burg, Hughes, Moshell & Lang, 1990; Pentland , 1990).

The interface dimension consists of work to further define and increase our understanding of how to develop complex, yet intuitive and natural controls to make virtual reality systems more responsive to human participation. Accordingly, we need to improve our understanding of human sensory mechanisms to be able to design and implement tactile feedback, binaural hearing, and autosterioscopic vision. We also need to improve our understanding of human perception for several reasons. As Zeltzer (1990) indicated:

> ... it is important to develop a taxonomy of tasks in terms of sensory input; for a given task, what sensory cues are necessary, and which cues are dispensable but improve performance? Are there sensory cues which do not affect performance per se, but which enhance the aesthetics of the operations or the work place? Are there sensory cues that interfere with performance and which should be avoided? (p. 5)

VIRTUAL REALITY AND TRAINING R&D

It would seem that a computer-based learning system that includes the user in a naturally participative and physically realistic setting would be more instructive than conventional computer-based systems. Similarly, the explosion of "high tech" media in the last two decades lured many trainers into mistakenly assuming that increased learning would result from improved delivery systems (Clark, 1991; Clark, 1983). Accordingly, an overemphasis on the more salient aspects of simulation and a lack of emphasis on instructional principles led to the development of simulation devices that produce compelling experiences but are not optimally effective for training (Andrews, 1988). Consequently, most researchers and developers have learned that mere exposure to

even the most motivating experiences do not guarantee the acquisition of knowledge or skill.

It is only natural for people to focus on virtual realities that are the most exciting and unusual. However, the most fruitful training research and development is likely to result from a systematic examination of military training activities that require an experiential approach. Three generic areas, defined by Wiggers et al (1989) and by Monet, et al. (1990), come readily to mind: combat mission training, mission preview, and mission rehearsal.

Combat Mission Training

According to Wiggers, et al. (1989), combat mission training occurs when "Tactical forces/crews conduct training scenarios, to which some factors, including a moderate level of uncertainty, have been realistically applied with the intent of training for a particular type of mission." The purpose of the training is to increase crew's effectiveness in a variety of situations and not just the one specifically represented by the training scenario. In such training situations, the mission, terrain and opposing force(s) can be generic rather than specific. It is irrelevant then, whether the computer database contains 'real' or generic terrain information since learning a specific scenario is not the training objective. Crew members view the scenario either through helmet mounted displays or within a dome. Movements (head and eye as well as hand and body) are detected and responded to appropriately by the system without discernible lag. The crew member can interact with other members of the team as they perform the simulated mission. Opposing forces, which can be either manually controlled, semi-automated, or fully automated are also included in the scenario. Individuals' actions usually occur in real time, but time may be compressed or expanded in order to enhance training effectiveness. (Knerr, 1991)

Mission Preview

Wiggers, et al. (1989) defined mission preview as "Tactical forces/crews conducting initial familiarization for a specific mission. This can be performed using personal computers or similar equipment." The purpose of such preview activities are (a) to develop and refine, through the use of a simulated environment, a plan for a specific mission, and (b) to insure that crew members understand their role in the plan (Knerr, 1991). Mission preview calls for crew members to understand when and where to perform, not necessarily how to perform. That is, the emphasis is on cognitive, rather than psychomotor aspects of mission performance. In such scenarios, the database should represent the actual terrain over which crew members will conduct their mission, and the opposing forces (and other aspects of the mission) should represent the actual situation. In addition, actions take place in real or faster-than-real time. Feedback should be designed to provide information necessary to improve the plan and does not necessarily need to be given to individual crew members. (Knerr, 1991)

Mission Rehearsal

Monette, et al. (1990) defines mission rehearsal as "Tactical forces/crews conducting trial performances, to which all factors ... have been realistically applied to a situation with the intent of preparing for a specific mission. Mission rehearsal requires that the simulation represent to the highest degree possible, the terrain, mission, and opposing force(s) of a specific situation. The training which occurs in such simulations is intended to directly transfer to a specific, real world situation. Crew members' actions occur and the simulation responds in real time. Feedback is used to to increase mission success, rather than improve generic combat skills. (Knerr, 1991)

In addition to the above mentioned experiential tasks, enhanced training research and development is likely to result from a systematic examination of learning domains where abstract information is made more "concrete" through virtual reality. Can, for example, air combat maneuvering techniques become more obvious if energy management can literally be seen or felt? And how "real" do virtual realities need to be to accomplish their purpose?

Nearly 20 years ago James Batter (as reported by Foley, 1989) noted that some students studying graphic displays of two-dimensional force fields gained a better understanding of the concepts involved if they could not only see the force vectors but also feel them. Batter's study, using a very simple, two-dimensional, force-feedback device, illustrates at least one way research can be conducted to determine the training value of making abstract information more concrete through virtual reality.

CONCLUSION

The description of virtual reality given above shows that there is no one technology that creates a virtual reality. Virtual realities are more than helmet mounted displays, sensor gloves, or robotic devices. They are, instead, a fundamentally new way of looking at and developing interactive, computer-based simulations. Now that we understand the virtual reality experience and the medium that provides it, it's time to get to the real task at hand. That is, to research, design, and develop virtual reality-based simulations which provide effective training for the end user.

REFERENCES

Alessi, S. M. (1988). Fidelity in the design of instructional simulations. Journal of Computer-Based Instruction, 15(2), 40-47.

Andrews, D. H. (1988). Relationships among simulators, training devices and learning: A behavioral view. Educational Technology, 28(1), 48-53.

Burg, J., Hughes, C. E., Lisle, C., Moshell, M., Carrington, J., & Xin, L. (1991, May). Behavioral Representation in Virtual Reality. In D. E. Mullally, M. Petty & S. Smith (Eds.), Second Behavioral Representation and Computer Generated Forces Symposium Proceedings. University of Central Florida, Orlando, Fl.

Burg, J., Hughes, C. E., Moshell, J. M., & Lang, S. D. (1990). Constraint-based programming: A survey (Tech. Rep. No. IST-TR-90-16). Orlando, FL: University of Central Florida, Institute for Simulation and Training.

Carrington, J., Hughes, C. E., Burg J., & Xin, L. (1991). Physical modeling: A review of current research (Tech. Rep. No IST-TR- 91-11). Orlando, FL: University of Central Florida, Institute for Simulation and Training.

Clark, R. E. (1983). Reconsidering research on learning from media. Review of Educational Research, 53, 445-459.

Clark, R. E. (1991). When researchers swim upstream: Reflections on an unpopular argument about learning from media. Educational Technology, 31(2), 34-40.

DeFanti, T. A., Brown, M. D., & McCormick, B. H. (1989). Visualization: Expanding scientific and engineering research opportunities. Computer, 22(6), 12-26.

Fisher, S. S. (1991). Virtual Environments: Personal Simulations & Telepresence. In S. K. Helsel & J. P. Roth (Eds.), Virtual reality: Theory, practice, and promise (pp. 101-110). Westport, CT: Meckler.

Foley, J. D. (1989). Interfaces for advanced computing. Scientific American, 257(4), 126-135.

Gonzalez, A. (1991, May). State-of-the-art in high Autonomy Systems. In D. E. Mullally, M. Petty & S. Smith (Eds.), Second Behavioral Representation and Computer Generated Forces Symposium Proceedings. University of Central Florida, Orlando, Fl.

Heim, M. (1991). The metaphysics of virtual reality. In S. K. Helsel & J. P. Roth (Eds.), Virtual reality: Theory, practice, and promise (pp. 27-34). Westport, CT: Meckler.

Helsel, S. K., & Roth, J. P. (Eds.). (1991) Virtual reality: Theory, practice, and promise. Westport, CT: Meckler.

Knerr, B. (1991). Behavioral requirements for training and rehearsal in virtual environments. Unpublished Manuscript, Army Research Institute for Behavioral and Social Science, Alexandria, VA.

Le H. T. (1991, May). Multiple autonomous combatants: Control and Navigation. In D. E. Mullally, M. Petty & S. Smith (Eds.), Second Behavioral Representation and Computer Generated Forces Symposium Proceedings. University of Central Florida, Orlando, Fl.

Lee, P., Wei, J., Zhao, J., & Bradler, N. (1990). Strength-guided motion. Computer Graphics, 22(4).

Malone, T. W. (1981) Toward a theory of intrinsically motivating instruction. Cognitive Science, 4, 333-369.

Monette, R., George, G., & Knight, S. (1990). Mission training and rehearsal employing simulation to its full potential. Proceedings of the 12th Interservice / Industry Training Systems Conference, 12, 409-413.

Moshell, M. (1990, May). Virtual reality: Revolutionary paradigm or yet another computer-hype? Unpublished presentation, University of Central Florida, Institute for Simulation and Training Orlando, FL.

Pentland, A. P. (1990). Computational complexity versus simulated environments. Computer Graphics, 24, 185-192.

Peterson, I. (1990). Recipes for Artificial Realities. Science News, 138, 328-329.

Rolfe, J. M. & Staples, J. (1986). Flight Simulation. Cambridge: Cambridge University Press.

Spring, M. B. (1991). Informating with virtual reality. In S. K. Helsel & J. P. Roth (Eds.), Virtual reality: Theory, practice, and promise (pp. 101-110). Westport, CT: Meckler.

Walsher, R. (1991). The emerging technology of Cyberspace. In S. K. Helsel & J. P. Roth (Eds.), Virtual reality: Theory, practice, and promise (pp. 101-110). Westport, CT: Meckler.

Wenzel, E. M. & Foster, S. H. (1990). Realtime digital synthesis of virtual acoustic environments. Proceedings of the 1990 Symposium on Interactive 3D Graphics (pp. 139-140). Snowbird, UT.

Wiggers, R., Hiteshew, L. K., & Matusof, R. (1989). Mission rehearsal: More than just another simulation. Proceedings of the 11th Interservice / Industry Training Systems Conference, 11, 569-573.

Wilhems, J., Moore, M., & Skinner, R. (1988). Dynamic animation: Interaction and control. The Visual Computer, 4, 283-295.

Zeltzer, D. (1990). Virtual environments: Where are we going? Paper presented at the 12th International IDATE (Institute de l'Audiovisuel Telecommunications en Europe) Conference, Montpellier, France.

ABOUT THE AUTHORS

Richard Thurman is a research psychologist at the Armstrong Laboratory, Aircrew Training Research Division. Holds a Ph.D. In instructional Science from Brigham Young University. His current research interests include instructional simulations, AI based training applications, and virtual environments.

Joseph Mattoon, also from Armstrong Laboratory, is currently a Ph.D. candidate in Educational Technology at Arizona State University. His research efforts are focused on instructional control mechanisms for training simulations and applications for virtual reality systems.

FORWARD LOOKING INFRARED SIMULATION
FIDELITY IN AIRCREW TRAINING DEVICES

Peter M. Crane
Armstrong Laboratory
Aircrew Training Research Division
Williams Air Force Base, Arizona

ABSTRACT

Simulating FLIR imagery requires integration of visual simulation technology with IR modeling and prediction. Analyses of simulated FLIR and consideration of training needs indicates that high fidelity simulation of all FLIR components is not required for many aircrew training applications. The different components of FLIR simulation i.e., predicting IR exitance from surface features, modeling atmospheric attenuation, and simulating sensor effects, are independent and an appropriate level of simulation complexity for each component must be selected for a particular application. Effective combinations of component fidelity for various training applications are described. Simulating sensor and atmospheric effects on FLIR imagery will have a high training payoff for many applications at relatively low cost. Developing a FLIR simulation system which will support thermally accurate IR predictions for any user specified mission scenario will require extensive development and data base support; the training applications which require such systems are limited.

INTRODUCTION

Forward Looking Infrared (FLIR) systems function by detecting the minute differences in infrared (IR) energy radiated from surfaces and imaging these differences as variations in brightness on a video display. Although a FLIR image gives the first impression of being very similar to a visual image, there are significant differences. The most important of these differences is that FLIR imagery varies over time. Due to changes in environmental conditions, FLIR imagery for a given scene will vary in visibility range, level of contrast, and the relative brightness of objects. A fully developed FLIR simulation would therefore incorporate the imaging capabilities of visual simulations with the ability to alter the thermal characteristics of the scene based on environmental conditions specified in training scenarios. Peters [4] estimated that 27 color tables would be necessary to simulate the different scenes resulting from variations in cloud cover, rain, wind speed, and IR viewing range for any given place, day, and time. Add color tables for effects due to air temperature, time of day, and month of the year and the number of combinations will easily exceed 10,000.

A very high fidelity FLIR simulation will include high quality visual imagery plus the capability to recreate the IR characteristics for any desired set of environmental conditions. However, the costs for IR effects modeling and database construction for this type of system will be very high. A more practical approach is to selectively incorporate different levels of fidelity based on training considerations: what is the intended use for the system, who will use it, how often, and what skills are to be learned? By applying the answers to these questions to the different components of a FLIR simulator for aircrew training, the designer can control system cost and complexity by limiting the range of variability to be simulated. This level of control is possible because there are three independent components within FLIR simulation: estimation of IR exitance from surfaces, atmospheric attenuation of IR energy between the surface and the sensor, and effects due to the sensor itself (Geltmacher [3]). Each of these components requires a separate modeling effort with its own advantages, costs, and limitations. Comparing the costs and benefits of simulating each of these components with the requirements for a particular training task allows designers to choose an appropriate level of simulation fidelity.

SIMULATING FLIR

IR simulation and modeling is a well-developed and mature technology most often used in systems development, targeting, and image analysis. High fidelity simulation for a small target area is the paramount concern. Most often, modeling is limited to a single target on a uniform background. Predictions for multiple objects within a scene or speed of computation have rarely been concerns for these types of simulations.

Simulating navigation and targeting FLIRs for aircrew training devices presents different problems from simulation for engineering analysis. Flight training simulators need to display a relatively large area within the sensor's field of view, in real time, and be compatible with available input data and computer image generators. Flight training systems must allow simulation of a variety of tactical situations with sufficient accuracy to train combat skills. Armstrong Laboratory has developed a FLIR simulation system based on integrating IR modeling and simulation with real-time visual simulation (Crane and Evans [1]).

Approach

The approach selected was to combine off line IR preprocessing, a real-time visual computer image generator (CIG), and sensor simulation postprocessing. The off line preprocessor function is to accept scenario and weather input and to estimate IR exitance for features in an existing visual data base. In real time, the CIG then converts exitance values into gray scale values, generates visual imagery for the specified field of view, adds texture, and determines visibility range based on atmospheric conditions. The postprocessor then simulates sensor functions, effects, and controls.

IR Energy Estimation

The preprocessor converts a visual data base into an IR data base. The visual data base consists of face definitions and

red-green-blue color codes assigned by the data base modeler. For input to the IR preprocessor, each face must also carry the Defense Mapping Agency (DMA) Feature Identification Descriptors (FID) and Surface Material Codes (SMC) from the original Digital Feature Analysis Data (DFAD) database. These codes identify the feature type and surface materials of the objects to be depicted in the simulation. FID and SMC codes were developed to support radar rather than IR simulation. While algorithms are available to predict radar reflectivity directly from these codes, additional data is necessary to predict IR exitance. Estimates of the IR significant properties of 640 FID/SMC combinations were compiled by the algorithm's designers and stored within the preprocessor. These properties include emissivity, reflectivity material thickness and composition, thermal conductivity and other factors which affect IR signature (Wolfe and Zissis [5]). The values assigned for these properties are the parameters used in the IR prediction equations. Specific scenario inputs are provided by the user. These are:

- -Latitude and longitude, time of day, and day of year
- -Air temperature (can be estimated if desired)
- -Humidity, wind speed, and haze
- -Rainfall rate and rain temperature
- -IR band desired: 3-5 or 8-14 microns.

IR exitance in watts/square meter is then estimated using a set of prediction equations based on first principles.

Real time functions

The visual CIG accepts input from the IR preprocessor and outputs textured, monochrome imagery with appropriate field of view and visibility range. This step incorporates both visual imaging and simulating atmospheric effects. Sensor effects and controls are simulated in real time by postprocessing functions. Each of these steps contributes to overall simulation fidelity and variability.

Texture and Scene Realism. Use of CIG texture patterns greatly increases scene realism but often interferes with presentation of estimated IR exitances. In simulated FLIR imagery at Armstrong Laboratory, the effects of texturing often overwhelmed the results of IR exitance prediction. For example, exitance values for a farm were predicted under different conditions so that the metal barn was either warmer or cooler than the wood house (see Figures 1 and 2). However, when these scenes were imaged, the nearby trees and fields which consist of simple polygons with overlying texture patterns did not change brightness. When the texture feature was disabled, the predicted IR differences were apparent but scene realism suffered dramatically. Integrating texture with FLIR simulation required the development of special texture patterns which did display differences in IR exitance without sacrificing scene realism.

Other special texture patterns can also increase the fidelity of simulated FLIR imagery. For example, desert plants such as creosote or tumbleweed show more contrast and texture in IR than in visible light. This effect can be simulated by using different texture patterns for visual and FLIR imagery. The resulting imagery displays both IR modeling effects and scene realism but with increased database development cost.

Figure 1. Simulated IR image of a wood house and metal barn with textured fields and trees.

Figure 2. Scene from Figure 1 imaged for different environmental conditions. Note that the predicted IR exitance for the house and barn have changed but that the brightness of the textured trees and fields have not.

Atmospheric Effects. The atmospheric effects which were modeled in the IR preprocessor only simulate effects of humidity or haze on the amount of sunlight which falls on surfaces. The preprocessor cannot simulate real-time effects resulting from the atmosphere between the sensor and target. A range-visibility function in the CIG attenuates visual image contrast with increasing range. For FLIR simulation, a computer model of atmospheric transmissivity such as LOWTRAN is used to establish an IR range-visibility function based on levels of humidity and haze specified in the chosen scenario. These scenario specific functions are substituted for the visible light attenuation functions to determine FLIR visibility range (see Figures 3 and 4).

Figure 3. Simulated IR scene with low humidity and long visibility range.

Figure 4. Simulated IR scene from Figure 3 but with reduced visibility due to increased humidity.

Sensor Characteristics. Simulated sensor effects and controls are added to the monochrome imagery by real-time post processor functions. Simulated pilot controls include: polarity (white hot/black hot) selection, gain, level, and magnification on a targeting channel. Sensor effects include AC restoration, system faults, blur, and noise. A sensor effect which is difficult to simulate using the three stage approach is a FLIR system's automatic gain control (AGC). If a given system has eight levels of brightness, the AGC will assign level zero to the coolest object within the sensor's instantaneous field of view and level seven to the warmest. The effect of AGC is to provide an image with a full range of contrast from white to black for any scene content. In three stage FLIR simulation, object brightness is proportional to predicted exitance for all features in the data base. Potentially, exitance values might be required for ice and a blast furnace. These exitance values are then mapped onto a limited range of brightness levels. If a simulated scene contains objects with a wide range of predicted exitance values such as water, soil, and concrete, the overall contrast within the scene will be acceptable. When simulating low level flight using a navigation FLIR, however, it was found that many scenes contain relatively homogenous features such as soil, rock, and sand. Differences in predicted IR were quite small and the contrast in simulated scenes was very low. Imagery tended to contain a narrow range of grey values without any black or white.

Simulating AGC increases image realism greatly. Different approaches to simulating AGC, however, illustrate the tradeoffs between system complexity and the range of conditions which can be simulated. For example, a given system may be used primarily for training low level navigation over largely unpopulated areas. In this case, few objects would be expected to be much warmer or cooler than the surrounding terrain. AGC can be simulated for this system by selecting a function to map IR exitance onto brightness which will emphasize the middle range of values. This will maximize contrast for areas such as deserts or forests but there will be little distinction among warm or cool objects. While the solution is inexpensive and appropriate for the intended application, the system could not be used for training target recognition within a city where there are many warm objects to be imaged. Alternatively, predicting IR exitance at frame rates for objects within the sensor's field of view rather than preprocessing the entire database or frame rate image processing will also simulate AGC. Either of these approaches requires additional computing power but will increase the range of scenes which can be simulated.

Advantages and Limitations

Any FLIR simulation system must depict a visual database in terms of IR exitance for at least one set of environmental conditions. The IR preprocessor at Armstrong Laboratory is an attempt to increase FLIR simulation fidelity by allowing instructors to select any environmental conditions for a particular scenario. There are, however, limitations in this approach. The IR estimation algorithms apply only to DFAD features. IR signatures of smaller objects such as vehicles or any features with internal heat sources must be derived from other sources. Also features unique to the IR spectrum including heat trails and exhaust plumes must be modeled separately. The IR exitance prediction model is very sensitive to the material parameters assigned to each feature. Changing the assumptions made regarding the characteristics of surfaces such as material thickness or thermal conductivity can significantly change the predicted IR exitance. In Armstrong Laboratory's FLIR simulation system, the thermal characteristics of each feature are based only on the DMA FID and SMC codes and the modeler's knowledge of their typical characteristics. If the characteristics of a given feature are different from the DMA feature or if the modeler's assumptions for a particular object are not correct, the IR exitance estimate can be significantly in error. It must also be recognized that even the most sophisticated IR modeling system cannot generate an exact prediction of what a FLIR scene will look like at a given time and date. FLIR imagery is highly susceptible to transient local weather effects including clouds and wind gusts. Predicted IR exitance values must therefore be treated as having large error tolerances.

The IR preprocessor supports two functions in generating FLIR databases: 1) thermal fidelity, and 2) the ability to simulate a given gaming area for a variety of environmental conditions. It is also possible to develop a high fidelity FLIR database for a single set of conditions using alternate sources of IR information. Using this approach, most of the IR significant variables such as day of the year, time of day, and air temperature would be constant for all simulations. Other variables, however, including IR visibility range, overall contrast, target to background contrast for non-DFAD features, and FLIR systems effects could be changed at will using other functions within the simulator. Selecting the number and type of IR conditions to be supported for a given simulator must be based on training needs.

FLIR SIMULATION, IR FIDELITY, AND TRAINING NEEDS

The advantage of the three stage approach to FLIR simulation is that it provides designers with many options. Selecting among these options determines the level of fidelity and the range of variables which will be simulated by a given system. Careful consideration of training objectives should drive the selection process. For example, objectives for preliminary training on the characteristics of FLIR imagery include:

- Differences between IR and visual imagery
- Scene characteristics unique to FLIR
- The effects of environmental variables on FLIR
- Hazards such as the near invisibility of wires in FLIR
- Target appearance for a variety of targets and conditions

This type of training is not specific to a single FLIR system and does not require interactive flight through a gaming area. Instead, the emphasis is on high thermal fidelity, multiple examples of scenes under different weather conditions, and examples of many different types of scenes and targets. For this application, simulation would not provide the necessary level of image detail or thermal fidelity. Instead, video presentation of recorded real world imagery would be a more appropriate medium to fulfil these particular training objectives.

Initial Weapons Systems Training

The focus of initial training on a FLIR equipped weapons system is on safe operation and learning to use the various systems. Training is limited to a period of several weeks using highly scripted scenarios. The objectives of initial training are typically:

-Learn operating procedures
--Functions and applications of system features
--Displays and controls
--Handoffs, coordination, and timing of events

-Integrate FLIR operation, navigation, and weapons employment
--Consolidate visual tasks (e.g., target acquisition) with systems tasks (e.g., navigation and weapons release)
--Sequences of operations
--Workload management

Examining these training objectives helps to define the fidelity requirements for the three components of FLIR simulation.

IR Energy Estimation. There is no need in initial weapons system training to illustrate the effects of varying environmental conditions on FLIR imagery. Given the objectives and the limited duration of training, it is highly unlikely that an instructor would use this training to demonstrate the effects of time of day or air temperature on image quality. An IR database with rank order correlation to real world imagery for a single set of environmental conditions will fully support initial training. Low fidelity FLIR simulation such as monochrome visual imagery may also be acceptable providing that subject matter experts have screened the imagery to remove conspicuous errors.

Atmospheric Attenuation. IR visibility is largely a function of humidity and haze and can be varied using the CIG's on-line range-visibility function. The effect of poor IR visibility is similar to driving in fog. Decreasing IR visibility during the later stages of training would reduce the time a student has available to perform a given task since objects would not be visible in the FLIR until they were at relatively close range. This function could also be used to tailor a training sortie to match local weather conditions. If the training curriculum calls for the student to practice an operation in the simulator and then in the aircraft, IR visibility in the simulator could be set to match predicted real world conditions. Overall, changing the atmospheric attenuation function provides limited variability in FLIR imagery at a significantly lower cost than manipulating the IR database. Although this feature may not be a firm requirement, inclusion of atmospheric attenuation effects could have high utility depending on the training syllabus.

Sensor Effects. Very high fidelity is required in simulating the effects, displays, and controls of the specific FLIR system being simulated. The effects of operator controls such as gain, level, and polarity reversal must be accurately modeled since learning to use these functions is a major objective of initial training. Modeling system faults is also required if students are to learn specific procedures for diagnosing and responding to systems failures.

Continuation Training

The primary application for thermally accurate simulated FLIR imagery is full mission training using an Operational Flight Trainer (OFT). In this application, students focus on consolidating their skills and tactical employment of their weapons system under many different conditions. While the duration of a particular training event is limited, students can expect to return to the OFT many times over several years. The objectives of continuation training are typically:

-Upgrade skills to employ new systems or weapons
-Operate under simulated threats
-Tactics development and evaluation
-Full mission training

IR Energy Estimation: IR exitance of features. The capability to model variability in IR exitance will be of greater value for continuation training than for initial training. Using this capability, training scenarios for journeyman and expert crews can be tailored to demonstrate the range of conditions which might be expected within an operational environment. Opportunities for such training include: 1) before a deployment, 2) for new crews entering an area, or 3) for repeated simulator sorties within an operational area as the seasons change. This capability will require the ability to model the major environmental variables including time of day, month of the year, air temperature, and cloud cover. High fidelity modeling of absolute temperature differences among objects is not required. The effects of simulated AGC plus operator controlled gain and level controls will support full system operation if the relative exitance of features in the database have rank order correlation with the real world.

IR Energy Estimation: Database Enhancements. A number of enhancements will be necessary to support a thermally accurate database for continuation training. IR signatures of DFAD features which have internal heat sources such as a power plant with cooling towers or a heated building cannot be modeled accurately using a preprocessor similar to the one at Armstrong Laboratory. Non DFAD objects including vehicles must be modeled separately. Models of moving vehicles may also include heat trails and exhaust plumes which are visible only in FLIR. In addition, local weather effects such as snow cover or bodies of water which may freeze in winter must be added to the IR prediction model. These database enhancements will be of greatest training value for students in continuation training who have mastered systems operations skills. These students will be focusing on integrating FLIR systems operation into combat missions. Training for tasks such as navigation within populated areas, target recognition and discrimination will benefit the most from these types of database enhancements. Unlike the IR signatures of DFAD objects which change in response to environmental variables, the IR signatures of

features with internal heat sources are relatively constant. Once modeled, they can be inserted into different scenarios without modification.

Atmospheric Attenuation. Modeling of variable range-visibility due to humidity and haze is required for continuation training. In addition, local atmospheric conditions such as smoke, fog or rain showers can be incorporated into the database as translucent texture patterns.

Sensor Effects. As for initial training, very high fidelity is required for modeling sensor effects, controls, and displays.

SUMMARY AND CONCLUSIONS

There are three components which must be evaluated when determining the fidelity requirements for a given FLIR training system. The first component involves the preparation of an IR database. Designers may elect to use fixed grey shades which represent a single set of environmental conditions. Alternatively, designers may choose to use a data base preprocessor which will predict IR exitance for any set of environmental conditions. The increases in flexibility provided by a preprocessor will be of significant importance for continuation training providing students will have the opportunity to train in the same gaming area under different conditions. The second component is the computer image generator which provides both imagery and a number of real-time FLIR effects. Simulating atmospheric attenuation of IR visibility resulting from haze and humidity demonstrates the variability of FLIR imagery at low cost. The third component is the simulation of FLIR system specific effects, displays, and controls. High fidelity is required for this component in all aircrew training devices.

Simulating FLIR imagery for aircrew training has been seen as the integration of visual simulation technology with IR modeling. The major emphasis within the science of IR modeling has been thermal accuracy. Due to the number of factors that influence IR exitance and the large number of assumptions that must be made regarding the IR characteristics of DFAD features, it has proven to be extremely difficult to incorporate high accuracy thermal modeling into aircrew training simulators. The other aspects of FLIR simulation, i.e., atmospheric and sensor effects, are much more tractable and will have a greater payoff for most training applications. By analyzing the training objectives of each proposed FLIR simulation system, it is possible to select the levels of fidelity required and to translate training requirements into engineering specifications.

REFERENCES

[1] Crane, P. M., and Evans, R. J. (1990). Thermal Accuracy in Simulated FLIR Imagery. Presented at the IMAGE V Conference, Phoenix, AZ.

[2] Evans, R. J., and Crane, P. M. (1989). Dynamic FLIR simulation in flight training research. SPIE Proceedings vol 1110, Imaging Infrared: Scene Simulation, Modeling, and Real Image Tracking. Bellingham, WA. SPIE--The International Society for Optical Engineering.

[3] Geltmacher, H. E. (1981). Weather and thermal electro-optical sensor performance. SPIE Proceedings vol 305, Atmospheric Effects on System Performance. Bellingham, WA. SPIE--The International Society for Optical Engineering.

[4] Peters, D. L. (1990). FLIR--A Different World. Presented at the IMAGE V Conference, Phoenix, AZ.

[5] Wolfe, W. L., and Zissis, G. J., eds (1985). The Infrared Handbook, Revised Edition. Ann Arbor, MI. Environmental Research Institute of Michigan.

ABOUT THE AUTHOR

Peter Crane is a Research Psychologist with the Armstrong Laboratory, Aircrew Training Research Division and is responsible for research on the training effectiveness requirements for simulating imaging sensors and electronic combat systems. Before moving to Arizona, Dr Crane was an Assistant Professor at the University of Pittsburgh at Johnstown (PA). During the summer of 1983 he was a visiting scientist at the Armstrong Aerospace Medical Research Laboratory. Peter Crane received his Ph.D in experimental psychology from Miami University (Ohio) with specializations in vision, cognition, and human factors.

RAPID-RESPONSE IMAGING SENSOR SIMULATION

Budimir Zvolanek
Erwin W. Baumann
McDonnell Douglas Training Systems
St. Louis, Missouri

ABSTRACT

Simulation of infrared, radar, and other imaging sensors plays an important role in the planning and rehearsal of military missions and in the training of mission personnel. The challenge is to develop technology that can use recently acquired intelligence information to quickly simulate cockpit sensor displays that accurately represent the real world while insuring correlation with the out-the-window displays and among the sensors.

This paper describes a novel, neural-network-based technique for infrared and radar image simulation directly from multi-spectral imagery. Source imagery, its processing using neural networks, and infrared and radar image simulation results are presented.

INTRODUCTION

The importance of imaging sensors, including infrared and radar, to the successful execution of navigation and air-to-ground targeting functions has been clearly demonstrated in practice. Such sensors could be applied even more effectively by mission personnel who have planned, previewed, and rehearsed the use of these imaging sensors prior to actual mission execution. To support this training process it is important that sensor imagery be simulated rapidly using the most recently acquired intelligence data.

Currently available image generators employed for sensor simulation use databases that describe mission gaming areas in terms of discrete features, three-dimensional models, and digital terrain elevations. Such information is derived from terrain photography, digital imagery, and a variety of cartographic products. Unfortunately, the process of reducing the original source imagery into a compressed set of attributed features and terrain models "filters out" most of the detailed spectral information available in the source imagery and also discards the desirable real-world appearance of the imagery. Because such databases are coarse representations of the real world, sensor images simulated from these databases also represent a coarse approximation to the real world.

Furthermore, although a semi-automatic process of extracting feature models and attributes from source imagery can produce highly detailed representations of selected areas of interest, this can be a very time-intensive and subjective process. Consequently, in situations requiring rapid database generation or updating, only the most critical areas can be modelled adequately to support high-fidelity sensor simulations. The remainder of the mission gaming area is still simulated at low fidelity using sparse and/or old data.

This paper addresses image-based techniques intended to overcome the limitations of traditional feature-driven sensor simulation approaches by directly transforming multi-spectral imagery into a sensor image. Such techniques have the potential to provide rapidly simulated, high-fidelity, sensor imagery correlated with other sensor imagery and the real world over large gaming areas.

After a brief introduction to image-to-image transformations and artificial neural networks, initial results in infrared and radar image intensity simulations are presented, followed by conclusions regarding future investigations.

IMAGE-TO-IMAGE TRANSFORMATIONS AND ARTIFICIAL NEURAL NETWORKS

Figure 1 compares traditional and image-to-image sensor image generation approaches. The image-to-image approach directly and in real time transforms multi-spectral imagery (MSI) from a run-time database into the desired sensor image. The run-time database is a mosaic of multiple resolution images reformatted and pre-processed off-line from real-world MSI. Both infrared and radar images, for example, are to be generated directly, at run-time, from the same multi-spectral database, ensuring correlated infrared and radar displays and an accurate representation of the real world.

In contrast, the traditional approach to sensor image generation requires an intensive analysis of MSI to develop models of real world features which are then inserted into a common database. The database is converted into either a mosaic of gridded data or polygons which are in turn used to simulate the sensor imagery. A major limitation of this approach is the substantial time and resources required to perform the necessary feature extraction and modelling needed to generate or modify the common and run-time databases.

The ability to perform successful image-to-image transformations for sensor simulation requires the selection of a set of multi-spectral input image channels that, in combination, contain sufficient information to predict the desired image waveband. Also needed is a way to deduce the inter-band relationships to be used for the image prediction/transformation function, and to apply that transform to the stored database images to produce the desired sensor image in real time. The technique presented here attempts to achieve this by applying the learning and recall capabilities of artificial neural networks to multi-spectral imagery.

Figure 1 illustrates infrared (IR), radar, and night-vision goggle (NVG) processors implemented using artificial neural networks (ANNs). Figure 2 shows a typical multi-layer, feed-forward neural network having three inputs, eight outputs, and three trainable

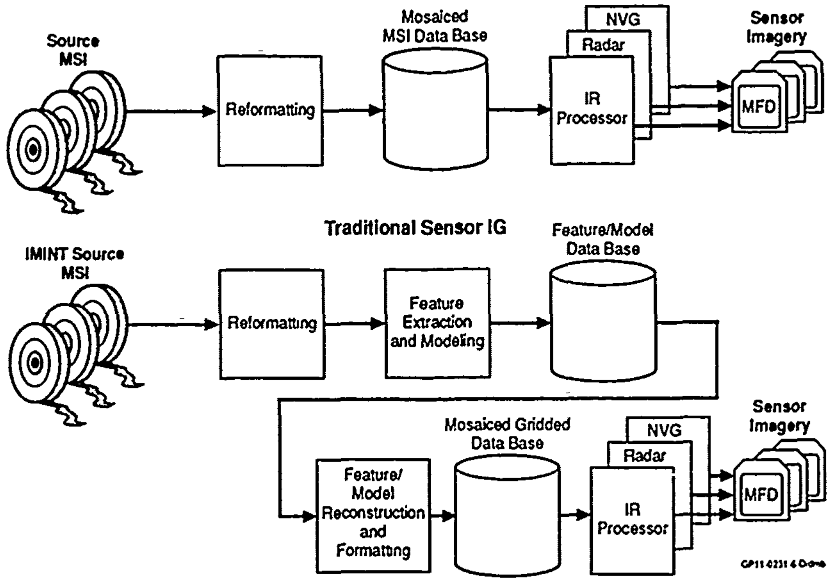


Figure 1. A Comparison of Image-to-Image and Traditional Sensor Simulation Approaches

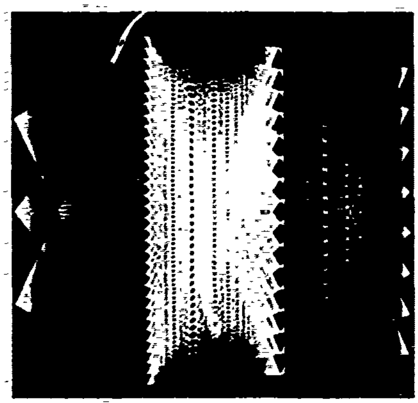


Figure 2. A 3-32-16-8 Multi-Layer Feed-Forward Artificial Neural Network

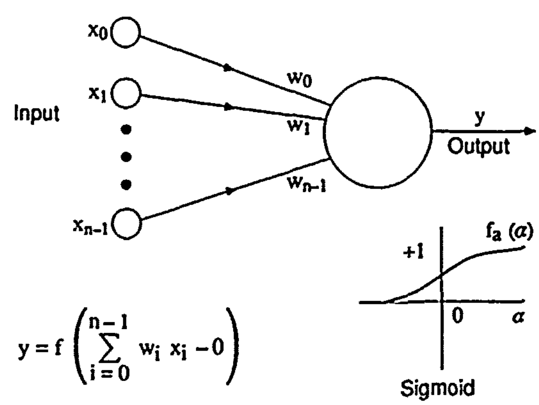


Figure 3. Neuron – The Basic Processing Element of Artificial Neural Networks

layers. Each layer contains many identical processing elements referred to as "neurons". A typical neuron, as shown in Figure 3, consists of a summation operator followed by a non-linear transfer, or "activation", function. Using a backpropagation-of-errors algorithm[1], the network is trained by iteratively presenting examples of input values paired with corresponding desired output data and adjusting the neuron input weights until optimal agreement between the predicted and desired output values is achieved.

Figure 4 illustrates the concept of training and applying neural networks for radar image prediction using MSI. During training, synthetic aperture radar (SAR) imagery is presented to the network's output while MSI of the same geographic area is simultaneously presented to the three network inputs. During production, MSI produced by the same or similar sensors is passed through the neural network to predict SAR image intensities.

INFRARED AND RADAR IMAGE SIMULATION RESULTS

We have performed two experiments to investigate the potential for direct transformation of MSI into IR and SAR imagery.

The first experiment involved training a back-propagation neural network (BPN) on Landsat Thematic Mapper (TM) MSI to predict a near-IR band from other TM MSI bands as an initial step toward predicting Forward-Looking IR (FLIR) sensor imagery. Multi-spectral pixels for network training were randomly selected from the region surrounding Washington, Missouri. Their values in TM bands 2 (0.52-0.60 um), 4 (0.76-0.90 um), and 5 (1.55- 1.75) were used to train a multi-layer feed-forward neural network to predict the near-IR band 7 (2.08-2.35 um). The network had three-inputs, two hidden layers consisting of thirty-two and sixteen nodes, and an eight-node output layer (256 intensity levels). It was trained using the back-propagation-of-errors technique by repetitively "showing" the network a training database containing 16,384 pixels randomly sampled from the Washington, Missouri, MSI database.

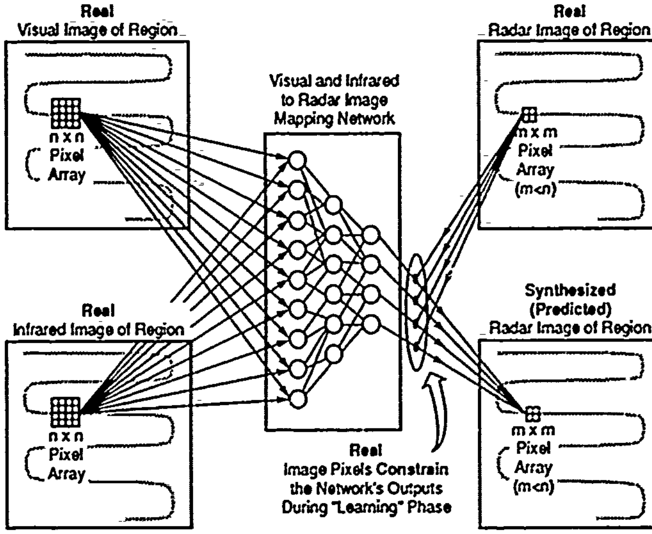


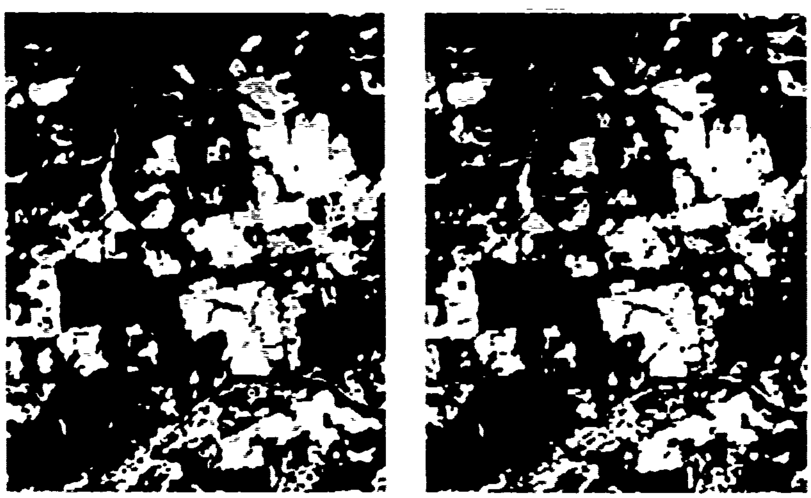
Figure 4. Training an Artificial Neural Network for Image Prediction

After ten passes through the training data, the network error had stabilized and the connection weights were stored. The trained network was then used to predict the near-IR TM band 7 for an area near Wentzville, Missouri, again using TM bands 2, 4, and 5 as input. The predicted pixel intensities were formed by converting the eight BPN outputs into an 8-bit, 256-grey-level pixel intensity. The results are shown in Figure 5 where the actual band 7 image is on the left and the predicted band 7 image is on the right. The differences in grey levels between the predicted and actual TM band 7 images were minor, indicating the network learned the multi-band intensity transformation quite well.

The long-wave IR TM band 6 (10.4–12.5 um) was also used for FLIR simulation experiments. However, the low spatial resolution (120 m/pixel) and coarse grey-level quantization (9 grey levels in the training pixels and 22 levels in the testing images) of TM band 6 contributed to the production of highly segmented images that were not good reproductions of the actual band 6 image.

In the second experiment a similar BPN network was trained to predict a SAR image of downtown St. Louis, MO. This time, a training database was created by manually selecting 400 pixels of SPOT satellite MSI data registered with a flight-test SAR image. The SPOT MSI bands 1 (0.50– 0.59 um), 2 (0.61–0.68 um), and 3 (0.79–0.89 um) and an X-band (10-GHz) radar were used. The training data included pixels taken from the river, a bridge, streets, highways, bare ground, grass, and several structures including a stadium, parking garage, and power plant.

After 1,400 iterations through the database, the network was used to predict the SAR image shown in Figure 6. Over fifty percent of a random set of pixels taken from the actual SAR image were found to be identical to those predicted by the neural network.



Actual Predicted

Figure 5. A Comparison of Actual and Predicted Landsat TM Near-IR Band 7 Images of Wentzville, Missouri

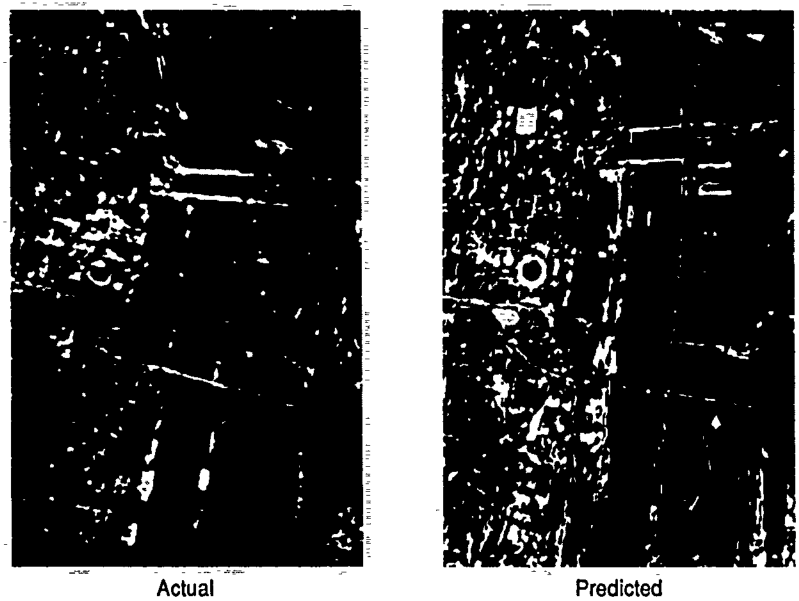


Figure 6. A Comparison of Actual and Predicted X-Band SAR Images

One result of particular interest is that the two northern bridges showed up clearly in the predicted image, but only the supporting piers of the southernmost bridge were highlighted. This was due to the fact that the northern bridge, from which training pixels v:ere taken, is primarily a stone and concrete structure, while the southernmost bridge is constructed almost entirely of dark metal. No pixels from the southern bridge were included in the origin il training database. As a result, only the concrete piers were "recognized" by the neural network and shown in the predicted radar image. This omission was resolved by adding six pixels from the upper portion of the "missing" bridge to the training database and retraining the network. The newly sampled bridge (and other spectrally similar objects) then appeared in subsequent predicted radar images.

Another interesting feature of the predicted SAR image is the presence of a bright, square structure in the upper left that is not present in the actual SAR image used to train the network. This is not a "stealth building", but instead is not visible in the actual radar image simply because the structure did not exist when the radar image was made but was present several years later when the SPOT satellite imagery was acquired.

CONCLUSION

The infrared and radar image prediction results achieved so far are promising but indicate the need for additional research. For example, although the neural-network predictions of near-IR images were very good, the far-IR imagery produced so far has been much less accurate. To what degree the low fidelity is due to the inherent difficulty of the prediction and how much it is caused by other factors remains unclear and requires further investigation.

The unexpected appearance of a recently completed building in the predicted SAR image of St. Louis (not present in the SAR imagery used to train the network) demonstrates one of the advantages of simulating sensor displays directly from the latest reconnaissance imagery. The predicted sensor imagery, even if not perfect in every respect, will be as up-to-date as the latest reconnaissance imagery, and will be automatically correlated with other sensor and visual displays also generated directly from the same MSI.

The results described here represent the initial phase of our investigation into the capabilities and limitations of this approach to imaging sensor simulation. Further research is required in many areas including limitations due to the effects of weather, Sun angle, clouds, and other atmospheric conditions on neural-network training and image prediction over a variety of geographic areas.

Experimentation is also needed to determine how to best generate and blend terrain-dependent 3-D effects into the predicted radar imagery. For example, no attempt was made to predict the "speckle noise" evident in real SAR images or 3-D effects such as shadowing and far-shore brightening.

Although our research has only begun to investigate the potential, and limitations, of a direct-from-MSI, image-to-image approach to sensor simulation, we believe that our neural network based techniques for learning multi-spectral image transformations will provide a powerful tool to support the rapid simulation of sensor imagery over large gaming areas.

REFERENCES AND SUGGESTED READING

[1] Rumelhart, D., and McClelland, J., "Parallel Distributed Processing, Explorations in the Microstructure of Cognition, Volume 1: Foundations", MIT Press, Cambridge, MA, 1986

[2] Aleksander, I., "Neural Computing Architectures", MIT Press, Cambridge, MA, 1989

ABOUT THE AUTHORS

Budimir Zvolanek is a Technical Specialist in Product Development at McDonnell Douglas Training Systems (MDTS) in St. Louis. Mr. Zvolanek has concentrated his career on the application and development of electronic imaging systems, image processing, digital signal processing, and computer-based data acquisition. Having conceived the image-to-image sensor simulation approach, he is currently responsible for research and development activities in sensor simulation and correlated databases. Previously, he lead the development and design of the F-14A radar simulator under the Navy 2E6 Tactical Scenario Improvement program at MDTS and the Video Image Dynamics System for the Standoff Land Attack Missile at McDonnell Aircraft Company Flight Simulation Laboratory. While at the McDonnell Douglas Missile Systems Company, he led the Advanced Anti-Ship Targeting Development effort to automatically recognize ship targets from infrared imagery and supported algorithm development for laser radar imaging. Mr. Zvolanek received his M.S.E.E. degree from Washington University in St. Louis, Missouri.

Erv Baumann has specialized in the application of artificial intelligence methods to sensor simulation, machine vision, and automated planning. As Technical Specialist at MDTS, he has conceived and developed innovative applications of neural networks to multi-spectral image analysis and sensor simulation. While working in MCAIR's Advanced CAD/CAM Technology group, he was responsible for the enhancement and application of an internally developed expert-system-shell used to implement a generative process planning system and several stand-alone expert systems. Prior to this, he assisted in the design and implementation of two model-based machine vision systems and developed sensor simulation software for an industrial robot simulation and programming workstation. Mr. Baumann received his B.S.E.E. degree from the University of Minnesota.

SENSOR DATA BASE CORRELATION

Dale H. Fawcett
Loral Defense Systems-Akron
Akron, Ohio

ABSTRACT

As modern aircraft become more dependent upon sensors, sensor correlation presents a growing challenge for modern mission rehearsal devices and multi-sensor training devices. The crew members are learning to perform full mission functions using a variety of sensors. These sensor displays must appear realistic and correlate correctly to provide for low level flight and sensor discrimination tasks. This is especially important in crew coordination tasks in mission rehearsal devices. The correlation problem can exist in training devices since sensor data bases are often procured from different vendors or generated from different source material. Technology limits of the image generators compound this problem by reducing the number of features that can be represented in the scene. Developers must construct sensor data bases carefully, with certain compromises, to assure realistic training while maintaining sufficient correlation and accuracy.

This paper describes how Loral is applying this critical technology, learned on the F-15E WST, to the Special Operations Forces (SOF) Aircrew Training System (ATS). Loral is generating a set of data bases to support visual, EO/IR, and various radar sensor simulations with a high degree of correlation. These data bases also meet a high accuracy specification to the digital, map, and photo source data, while being produced in only 48 hours. In addition, the interactive threat simulation entities correlate with all of these data bases. The result is highly realistic training and mission rehearsal devices which overcome the sensor correlation problems.

INTRODUCTION

Modern flight simulators are designed to train aircrews to perform their tasks efficiently. They have the advantage of being safer than flying and they provide a more rigorously controlled environment. They have to create very realistically simulated conditions so the transfer of training to the real airplane can occur easily. One of the more difficult areas to simulate realistically is the area of sensors. Sensors include various radars, infrared, electro-optical systems, and electronic warfare displays. Since the crew members use several of these sensor devices along with out-the-window visual cues while flying the aircraft, the simulator must provide the same realistic cues. These sensor displays must also have a high degree of correlation with each other to provide correct, realistic training to the crew members.

HISTORY

The early flight simulators included visual and motion systems, which along with the instruments provided the pilots with all their cues. It was quickly learned that these sensory inputs had to correlate or bad results such as simulator sickness occurred. As aircraft added more sophisticated equipment, additional requirements were added to the simulators. Using radar while flying at high altitude did not present significant problems in sensor correlation because of the coarseness of the display and the longer range of the radar compared to the visual. The radar image resembled the visual scene but did not correlate strongly for most trainer functions. As aircraft electronics improved, the challenge became greater. The use of terrain following radar allowed aircraft to fly safely at much lower altitudes. This

made the simulator correlation between visual and terrain following radar more critical for the pilot. These two inputs provide simultaneous displays of the same area; thus, a much better correlation was required. When we add to a single aircraft additional sensors of ground mapping radar with a high resolution synthetic aperture mode plus various infrared and electro-optical sensors, the simulation task becomes enormous.

BASIC PROBLEM

The training task in a multi-sensor aircraft becomes more complicated because it involves the synchronous use of more than one sensor at a time. It may also involve different tasks by several crew members coordinating with each other. For example, a pilot, while flying at low altitude, receives sensory inputs from the visual scene in front of him, from the terrain following radar display, and from the forward looking infrared (FLIR) display. These must all correlate so that he learns to use all of them together to perform his difficult task successfully. Other multiple sensor tasks are navigation updates and targeting. A synthetic aperture radar (SAR) is used for the long distance image of an area on the ground. As the aircraft approaches, an infrared sensor is slewed by the on-board avionics to the same location on the ground and provides the crew member with a new, closer range image. The navigation points often will be one unique building in a group of other buildings or some other cultural feature which is easily identifiable. These two images must correlate closely so the crew member can identify the selected object in the SAR image, and also identify the same features when he slews the infrared sensor to the given location. Now we add the last ingredient to the problem. All the crew members expect that all of these sensor displays will be realistic looking. They must look very much like they actually do in the aircraft. This is more than a desire, but actually a valid requirement for tasks such as target discrimination. This again is necessary for good transfer of training to the real flying environment. The simulation of these functions in a trainer requires a cost benefit trade-off. The simulator must provide sufficient simulation to train the crew members but at a realistic cost. Thus, certain compromises must be made along the way. Flying real aircraft provides excellent training, but it is very expensive and not all situations can be experienced as they can in a simulator.

F-15E EXPERIENCE

Loral encountered the challenge of sensor correlation with the F-15E Weapon System Trainer (WST). The F-15E aircraft is equipped with a real beam and synthetic aperture radar, a FLIR, a terrain following radar, a remote map reader, an infrared targeting pod, and electro-optical (EO) and infrared (IR) video from missiles. The F-15E aircraft flies at low altitudes and uses these sensors either simultaneously or sequentially in the operations. The F-15E WST simulates all of these systems plus the radar altimeter by means of correlated data bases. The result is a full system of many sensors providing the crew members with correlated displays. The F-15E WST is designed to meet a specification requiring that the radar data base be accurately generated with respect to its source data. This source data consists of Defense Mapping Agency Digital Feature Analysis Data (DFAD) and Digital Terrain Elevation Data (DTED) products which are enhanced by the addition of higher resolution data from map sources. The resulting data is quality checked to eliminate boundary inconsistencies and other possible anomalies. The radar data base has to match this source data with accuracies of up to 15 feet for point features in the target areas. Other accuracies are shown in Table 1. The IR and EO systems have their video generated by an Evans & Sutherland CT6 image generator. This system uses a standard polygonal data base but the resultant video has to correlate to the radar so that differences are imperceptible to the crew members. If the Digital Radar Landmass Simulator (DRLMS) data base only had to meet the accuracy specification, it could have been produced directly from the DFAD and DTED data, but it would not have correlated to the EO/IR system.

Polygonal data bases are used for visual systems because of time and memory constraints of the hardware. Providing a unique, real-time image of a large geographical area is not cost effective because of limited hardware technology. Using polygons for the terrain, with a set of features such as trees on them (called basis sets), allows the system to represent the large areas with relatively accurate terrain. The CT6 system

Table 1 — F-15E Radar Data Base

Accuracies	
Feature Type	Accuracy to Source Data (feet)
Target Areas	
Point	15
Lineal	15
Small Areal	15
Large Areal	64
Background Area	
Point	60
Lineal	150
Small Areal	150
Large Areal	1200

used on the F-15E WST employs a set of polygonal basis sets for the terrain with unique features placed on top as required. Each of the terrain basis set elements is an 800-meter equilateral triangle. It has various fixed features associated with it such as trees for a forest, cactus for desert, and buildings for cities. There are a limited number of these basis set elements possible in the system, so other features used to represent the real world features can be included. This data base was designed so that the terrain triangles all have vertices that are increments of 50 meters in height from the data base origin. The result is a set of triangles that match when positioned adjacent to each other, and form a good visual scene.

The following method was used to produce this data base. The DTED data is scanned and analyzed to get the best curve fit plane for each triangular area. Then the three vertices are adjusted to the closest 50-meter altitude. In addition, the city areas are given some variety by using different patterns for the roads and buildings. This adds realism to the scene by eliminating distracting repetitive patterns on the displays. The real cultural features are then located onto the terrain triangles. A limitation in this process is that the real features cannot cross the triangle boundaries as this may cause visual anomalies in the visual image. A correlated radar data base must therefore be made using the CT6 data base features so that only the same features in the same locations are placed into the radar data base.

Initial investigations, which used polygonal data bases for the radar data base, indicated that, though correlation was good, the image lacked realism because the polygons could be clearly seen. Figure 1 shows an example of a ground mapped radar image using a completely polygonal terrain data base generated by the F-15E DRLMS. It was therefore necessary to change the radar data base to make more acceptable images while still maintaining the correlation to the other sensor data base. This was accomplished by modulating the polygonal terrain data base before the 50-meter adjustment to the vertices with the source terrain data from the DTED tape. The result was a radar data base that generated acceptable images, met the accuracy specification, and correlated to the IR and EO sensors. The modulation process consists of converting the terrain triangle back into data posts spaced to coincide with the DTED data, then performing a weighted average with the original DTED posts. Figure 2 presents a radar image using this modulated terrain data base. Figure 3 represents a radar image from a pure DTED terrain without having been polygonalized. It can be seen that the differences from the Figure 3 image do not detract from the realism of the radar image generated from the combined polygon and DTED.

Terrain following radar (TFR) presented a similar problem; the image had to look realistic for cockpit displays and still had to correlate with the ground map radar and the FLIR. The FLIR correlation became the predominant driving force since it and the terrain following radar are used simultaneously while flying low to the ground. The specification on the TFR to FLIR correlation reflects this need as shown in Table 2. The terrain following data base was generated by gridding the polygons after they had been changed in elevation to move the vertices to the 50-meter altitudes. This method achieved a realistic image while maintaining very high correlation to the FLIR. This solution resulted in an acceptable compromise, accomplishing correlation and fidelity of TFR image sufficiently to support training.

Another function of the F-15E aircraft is sensor handoff from one system to another. The radar is used to generate a synthetic aperture radar image at a long range. This image shows sufficient detail to identify specific features. The crew member

Figure 1 — Ground map radar image of polygonal terrain data base

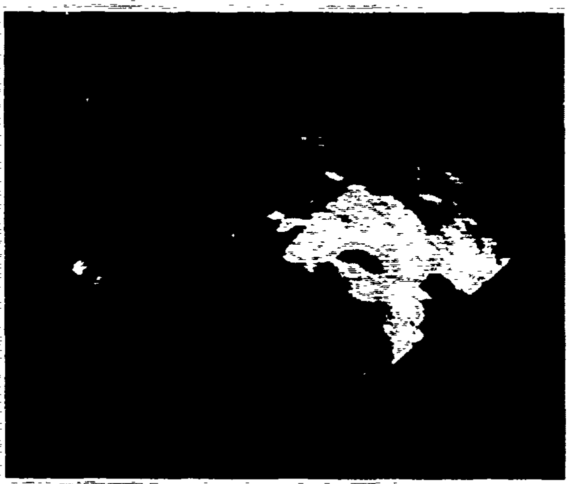


Figure 2 — Ground map radar image of DTED modulated terrain data base

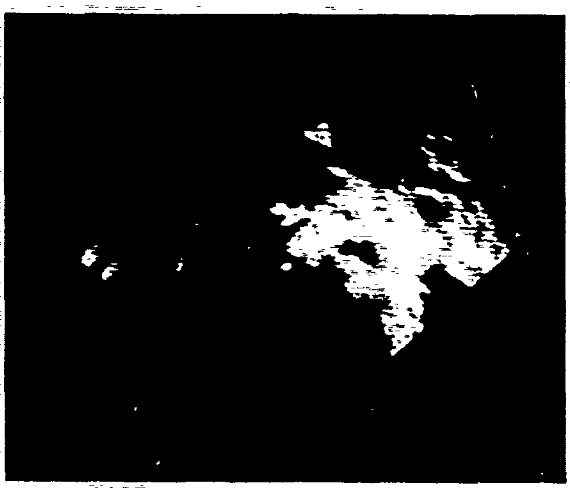


Figure 3 — Ground map radar image of DTED terrain data base

Table 2 — DRLMS to EO/IR Correlation

Specification Criteria	Ground Map to EO/IR	TFR to EO/IR
Mean	55 + 3R	10 + .5R
Standard Deviation	35 + 2R	30 + R
Maximum Deviation	100 + 25R	60 + 4R

Where R is the roughness index of the 61 x 61 points of terrain being compared.

then centers his cursor on a selected feature and saves the geographical location. As the aircraft approaches closer to the area, the crew member uses that saved information to slew his IR sensor via his avionics controls to the same location. He can then see a closer range image of the same scene. In the WST, this same function must be accomplished with a correlated image so the crew member learns the relationships between his equipments and their displays. When the selected target area is near a city, even the synthetic breakup areas must match, since the association of the correct target may be found by its relationship to adjacent synthetic features. The basis sets of the IR data base are used as models in the radar data base generation, so the synthetic features are located in the same relative positions in both data bases.

The F-15E sensor data bases provide a realistic image for the radars and IR/EO sensors that are simulated. These data bases and their images correlate and meet the accuracy specifications to the source data. Thus, they are providing a good training system.

THE SOF ATS CHALLENGE

The Special Operations Forces Aircrew Training System (SOF ATS) provides both WSTs for training full aircrews and Mission Rehearsal Devices (MRDs) for allowing fully trained crews to rehearse missions that are to be flown in the near future. This mission rehearsal requirement dictates that the data bases must be extremely accurate along with having the full sensor correlation. The mission rehearsal device is designed to support an aircrew in performing their critical functions in

the environment in which they expect to be flying their mission. It must provide highly realistic visual, sensor, and environmental cues to the crew members. They are preparing for an actual mission and this realism could mean the difference between success and failure of their missions.

The SOF ATS contract calls for the simulation of the Talon I and Talon II aircraft, with time-phased options adding helicopters, gunships, and a tanker. The Talon aircraft are basically C-130 airplanes used for transporting cargo or personnel. However, these airplanes have been modified with the addition of special sensors and other equipment to enable them to fly low level flights at night. As a result, the simulation of the full airplane presents a larger challenge than the F-15E, since it involves more crew members using more equipment that needs correlation.

The Talon airplanes are equipped with ground mapping radar, terrain following radar, forward looking infrared, radar altimeters, and many electronic warfare (EW) systems and displays. The crew consists of pilot, co-pilot, flight engineer, navigator(s), electronic warfare officer, and communications officer. In the later options of additional simulators for the gunships, there are additional crew member positions which have TV systems and other equipment. The purpose of the mission rehearsal facility is to provide a good mission rehearsal capability for these crews. The critical element of mission rehearsal is to see and hear the same things in the device as the crew will see in the real world. This means that the MRD must provide a good out-the-window visual scene along with the accurate sensor displays that correlate with the real world and with each other. To meet this challenge, Loral is building a Data Base Generation System (DBGS) to make visual, infrared, and radar data bases that will be both highly accurate and have a high degree of correlation. Adding to this difficulty is the time requirement to produce these data bases in 48 hours.

The MRD will use the new Loral DRLMS for the ground map and TFR simulations and the Evans & Sutherland ESIG-4000 Image Generator for the visual and infrared simulations. However, along with having to simulate these systems, the SOF ATS systems must provide an accurate simulation of the electronic warfare environment. The Talon aircraft are required to fly in dangerous areas to perform their missions. They attempt to do so by avoiding contact with the enemy. Thus, the WST and MRD must provide them with a realistic threat environment so that they can train and rehearse the tactics that allow them to do this. This threat environment must also properly correlate with the visual and radar data bases. The navigator makes much simultaneous use of his real world maps and his IR and radar displays while directing the pilot where to fly. Therefore, the data bases must match those maps very well. In the SOF ATS WSTs and MRDs, the threat and radio environment must be accurately simulated with respect to the real-world terrain. It may be critical in the MRD to know at what altitude you can fly and not be detected, so the real mission is a success. Another requirement of SOF ATS is to link two or more MRDs together so they can fly coordinated missions, detecting each other with their sensors and interacting as they would in the real world. These MRDs must have identical data bases so that the crew members see the same features since they are in audio communication with each other.

To achieve all of this, a fully coordinated approach to sensor simuation is required. In addition to the radar and visual data base coordination, the threat and weather environment must be integrated into this approach. The MRD simulation makes use of the DRLMS and visual data bases to assure that the threat environment is fully correlated and realistic. The real-world threat environment is volatile, with last minute changes possible in the location of the threats. Therefore, the physical data bases of terrain and permanent features are processed separately from the electronic environment to make the DRLMS and visual data bases. The Electronic Order of Battle information for the threat environment and weather information are added at the start of the simulated mission. During a Plan Mode, the various threats are placed in the data base, based upon their known locations. They are placed in the visual, radar, and electronic data bases at the same locations so all sensor systems correlate. A visual check is made in the Plan Mode to assure there will not be any visual anomalies at these locations. This is possible since differing source data may not match perfectly; a SAM site location from one

source may overlap the edge of a lake from another source. A minor correction to the SAM site location would be made in Plan Mode to assure that these anomalies do not appear to the crew members in the rehearsal.

During the mission, the threat simulation software controls the actions of the threat entities. These entities perform as they would in the real world in a linked command and control structure. The early warning radars search for any incoming aircraft and signal the tracking or intercepting systems to perform their functions. The simulation uses the DRLMS system to compute the line-of-sight capabilities of these systems and assure that they will only see the ownship aircraft when it is not occulted by terrain or cultural features. This occulting function is performed by giving the DRLMS system the two locations; it then verifies that no obstructions block the view. When the view is unobstructed, the threat entity begins to perform its normal functions. In the case where very near range occulting must be performed with great accuracy, the visual system is used. The two locations are sent to the ESIG-4000 and the line of sight is verified to not be obstructed. This function is needed because the resolution of the DRLMS data base does not allow this fine discrimination for very close ranges.

The ownship sensors also react and provide the crew members with the appropriate outputs. These outputs may consist of audio or visual display warnings that the ownship is being detected. Likewise, the radio communications are processed to assure that they can only be heard when they could be heard in the real world, so that they are not obstructed. Again, the DRLMS is used to verify that threats and radios would be detected by the ownship and not occulted by the terrain. If there is a limited obstruction, as detected by the DRLMS, the radio simulation will introduce noise to simulate the real world transmission difficulties. This becomes especially important for multiple aircrew communications when more than one MRD are rehearsing together.

The generation of the visual and DRLMS data bases for the SOF ATS is handled by the DBGS. It processes maps, photographs, and digital data of various sorts, including DFAD, DTED, and Digital Chart of the World (DCW). All of these products are converted into an intermediate data base which contains terrain, cultural features, and texture information. This data base is then simultaneously processed by the visual and radar post-processing software. The visual system software converts the data to the ESIG-4000 system data base. This state-of-the-art system maintains a separate terrain and cultural data base on line, thus eliminating the need to convert all terrain into large polygons. The result is a much more accurate data base. The photo texture capability allows for maintaining many more features than previous systems, since fewer polygons are required to simulate these features. As a result of these improvements, the radar data base can be processed directly from the same intermediate data base instead of the resultant visual one. For areas that use synthetic breakup, the same algorithm is used to place the synthetic features in both the visual and radar data bases so that they correlate in the final products. The ground map and TFR data bases can be the same ones, thus saving disc space in the DRLMS system and data base generation processing time. Where the F-15E DRLMS had a separate TFR data base with terrain and feature height only, the S DRLMS uses one common data base, with the TFR processing using only the part that it needs, i.e., terrain and feature heights.

Another area of sensor correlation involves weather simulation. The aircraft are equipped with a weather mode in the radars so they can detect severe weather conditions and avoid them. The proper weather effects have to be simulated in the visual, radar, and threat environment. The Plan Mode accommodates the updating of weather data from recent meteorological reports. This weather data is put into a global weather data base for the whole flying area. As the crew flies along, they may encounter different weather conditions. These same conditions are simulated in the other MRDs so all crews experience the same effects in the correct geographical areas. This single weather simulation also provides the parameters needed to simulate the weather effects on the radio, radar, and electro-optical transmissions. Thus, the realism is maintained for consistent effects in all the sensors. Weather conditions and fronts are simulated in the weather radar and the visual systems that are seen directly

by the crew members. The weather effects on other sensors such as EW and radio are correlated by having one global weather simulation.

CONCLUSION

Aircraft sensor simulation is improving in its accuracy and correlation through the use of new, improved hardware and software systems. The demand for additional sophisticated functions continues to increase as the technology develops to accommodate it. Thus, through the growing technology of improved hardware and sophisticated software, the correlated sensor simulation of SOF ATS has progressed from that of the F-15E. The WSTs and MRDs of the SOF ATS will be completed in early 1993 with the inclusion of the fully integrated and correlated sensor simulations for coordinated multiple crew rehearsals.

ABOUT THE AUTHOR

Dale Fawcett is working as the System Engineering Manager for the Special Operations Forces (SOF) Aircrew Training Sytsem (ATS) program at Loral Defense Systems - Akron. He is responsible for system engineering on the Talon I and Talon II Weapon System Trainers (WSTs) and Mission Rehearsal Devices. Mr. Fawcett has previously worked as the Software Project Engineer on F-15E WST. In that position, he was responsible for the development and integration of the operational and support software and for the EO/IR and radar data bases for the F-15E WST. He has also worked on the R&D development of the Loral DRLMS and previously on the Reference Scene Generation Facility for the Army Pershing II Missile System. Mr. Fawcett holds a BS in Mathematics from Bucknell University and a MS in Systems Management from the University of Southern California.

TRAINING IN BATTLEFIELD OBSCURANTS

Rudolph R. Gammarino
Loral Electro-Optical Systems
Pomona, California

James W. Surhigh
U.S. Army, PM TRADE
Orlando, Florida

ABSTRACT

A laser training system entitled Shoot Through Obscuration MILES (STOM) is being developed to operate with the Forward Looking InfraRed (FLIR) system during battlefield exercises. The STOM system is capable of ranges in excess of 6 km and can penetrate battlefield obscurants such as fog-oil, smoke, and dust. It is designed to complement the existing Multiple Integrated Laser Engagement System (MILES), which cannot successfully penetrate obscurants that limit visibility but can be penetrated by the FLIR.

STOM employs an rf excited CO_2 laser which operates in the center of the FLIR's spectral window at 10.6 μm. The laser is sealed, non-cooled, and can generate 9 mJ laser pulses at relatively high repetition rates consistent with laser safety requirements. The STOM system uses a newly developed non-cooled pyroelectric detector receiver.

A prototype STOM system has been tested with various battlefield obscurants through which hit(s) can be obtained on targets that are visually obscured but can be seen with a FLIR.

INTRODUCTION

The MILES system uses laser bullets to simulate the lethality and realism of the modern tactical battlefield. Eye-safe Gallium Arsenide (GaAs) laser transmitters, capable of shooting pulses of coded infrared energy, simulate the effects of live ammunition. The transmitters are easily attached to and removed from hand-carried and vehicle-mounted direct fire weapons. Detectors located on opposing force troops and vehicles receive the coded laser pulses. A MILES decoder then determines whether the target was hit by a weapon which could cause damage (hierarchy of weapons effect) and whether the laser bullet was accurate enough to cause a casualty. The target vehicles or troops are made instantly aware of the accuracy of the shot by means of audio alarms and visual displays, which can indicate either a hit or a near miss. MILES is used by the armed forces of the U.S. and many foreign governments.

MILES complements the abilities of the unaided human eye. The semiconductor lasers used in MILES transmitters emit at a wavelength very close to the eye's response, thus, if a target is obscured, MILES laser transmitters cannot penetrate the obscurant. Therefore, a gunner using standard MILES in a training exercise in an obscured battlefield is not able to "shoot" what he can see through his FLIR. This happens because light scattering is a function of the ratio of the particle size to the light's wavelength. The FLIR's emission is normally from 8 μm to 12μm, which is a wavelength that is 10 times longer than the standard MILES lasers. The STOM system complements target acquisition FLIR systems during battlefield exercises where visibility is impaired. Since the STOM's laser emission of 10 μm is in the center of the FLIR's window, the STOM laser can penetrate through the same obscurants that FLIR can see through.

STOM LASER TRANSMISSION SYSTEM

The STOM laser transmitter assembly is comprised of two components:

- STOM laser transmitter
- STOM laser controller

The STOM laser transmitter assembly consists of a CO_2 laser and collimating optics. This equipment and a boresighting scope are mounted on a base plate (Figure 1). This assembly fires the laser "shot" at a target.

The laser controller assembly (Figure 2) contains a Central Processing Unit (CPU) board, rf laser driver, RS232 communication port, and five 6V gel cell batteries. The assembly, which interfaces with the laser, is used to control the laser transmitter.

Figure 1. STOM Laser

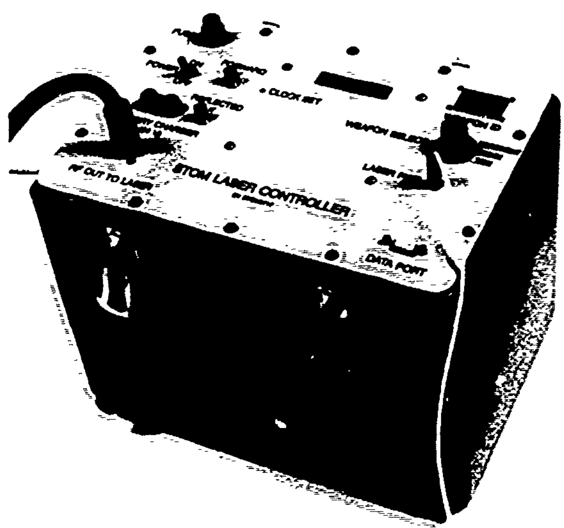


Figure 2. STOM Laser Controller

PRESENT STOM CO_2 LASER CHARACTERISTICS

The laser transmitter in the STOM system consists of a waveguide CO_2 laser, which is rf pumped at 150 MHz. The output wavelength is 10.59 μm. This folded-cavity laser is approximately 6 inches in diameter and has an equivalent cavity length of 1 m. The maximum output power of the laser is 30 W when excited with approximately 400 W of rf power. This design is not only 2 to 3 times smaller than traditional designs of comparable performance, but it is also more rugged and less susceptible to the thermal and mechanical deformations encountered in harsh environments.

The laser transmitter was tested at repetition rates as high as 6 KHz with 50 μs pulse widths. However, during the actual field testing and system analysis, a pulse width of 300 μs was used. This wider pulse width was required to obtain a high signal to noise ratio at longer ranges.

In order to obtain a nearly constant hit zone area independent of range, the laser beam was further collimated. A custom two-lens Galilean telescope was designed to obtain the desired laser beam patterns. The assembly is 6 inches long by 2 inches wide. The collimating optical assembly is extremely rugged and can be mounted easily to the laser transmitter assembly housing with four screws. A mounting plate was designed and built to hold the CO_2 laser, collimating optical assembly, and boresighting scope (Figure 1). The collimating optical assembly was mounted on the optical platform in front of the CO_2 laser and aligned to the laser output port.

A laser controller assembly was designed to incorporate all necessary components to run the laser and to generate the STOM codes and supply power for portable operation. The laser controller assembly is portable and completely self-contained. It needs no external power for operation.

A modified MILES II CPU board is used for encoding the various Player Identification (PID) and weapon codes, controlling the CO_2 laser, running the LCD display, storing data events, and running the RS 232 data link. The CPU board also contains a real time clock and a battery backed up RAM which can store up to 1000 events.

A boresight code is used to align optics and perform various tests. In this mode, the laser is fired at a rate of one pulse per second. In addition, either a TOW or 105mm code may be chosen. One of these codes is generated each time the fire button is pushed.

The STOM laser is powered by an rf oscillator/amplifier unit (rf unit) that supplies approximately 420 W of power at 65% efficiency when operated at 28 V dc. The power requirements for the laser and rf electronics can easily be met by using vehicle power. An average current of 3 A is required during a message transmission, and the idle current is expected to be below 50 mA.

A preliminary design of a STOM laser assembly for the M1/M1A1 tank is shown in Figure 3. This transmitter can be mounted in both the 105 mm and 120 mm main tank guns. The laser transmitter is nearly the same size as the present MILES 105mm tank transmitter. The alignment scope is not shown, but it will be mounted under the laser in the same manner as the standard MILES transmitter. The laser transmitter assembly will be attached to the laser controller via a 1 m rf cable. The laser controller will be placed on the tank floor, near the main gun.

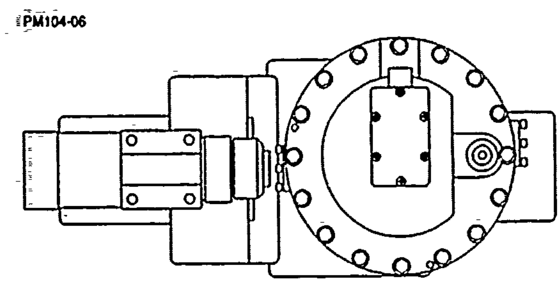


Figure 3. M1/M1A1 Tank Transmitter

RECEIVER SYSTEM

The STOM laser receiver system consists of two major components:

- Pyroelectric detector modules
- Decoder box (with optional strobe)

Up to 10 pyroelectric detector modules are connected in series with the decoder box. This decoder box is powered by four internally mounted 6 V gel cells. A MILES II CPU board with a receiver buffer board is used to power the detector modules and decode the incoming signals.

The detection system can be used in a stand-alone mode or as part of an M1/M1A1 tank system. In the stand-alone mode, the detection system is called the Mobile Independent Target System (MITS) (Figure 4). When configured as a MITS, the optional strobe light is connected/attached to the decoder box. As part of an M1/M1A1 system, the STOM detection system components are integrated into the existing tank system.

Two types of pyroelectric detectors were investigated: lithium tantalate and PVDF. Lithium tantalate is relatively expensive (estimated production cost \$200), whereas PVDF is a low-cost plastic material (estimated production cost of less than \$35). The preliminary results obtained with commercial PVDF material indicate that a 0.006 μm thick 1 cm^2 detector will have approximately the same responsivity as a 25 μm thick 1 cm^2 lithium tantalate detector. The coatings on the PVDF material tested were not optimized, so the actual responsivity measured was about 25% of the lithium tantalate detectors.

A low-noise hybrid amplifier was designed for use with the 1 cm^2, 25 μm thick lithium tantalate detectors. This amplifier is matched to the impedance of the detector and can detect as little as 100 electrons generated by the pyroelectric detector.

A surface-mounted Printed Wiring Board Assembly (PWBA) was designed to filter and further amplify the signal from the low-noise hybrid amplifier. This conditioned signal was then put into a threshold circuit that generated negative 5 V pulses when the incoming laser light level exceeded a preset value. Figure 5 shows a schematic of the actual STOM detector assembly. An Ar-coated Ge window is placed in front of the detector to protect it from the environment and filter out undesirable radiation. The detector is then placed in a module assembly, as shown in Figure 6.

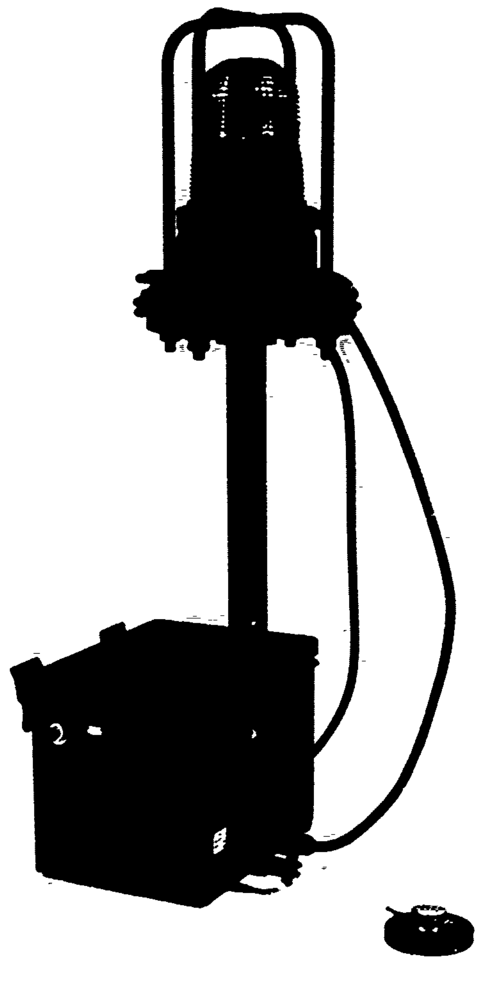


Figure 4. MITS

DECODER BOX

The STOM decoder box consists of the following components:

- Housing
- Receiver CPU board
- Combat Vehicle Kill Indicator (CVKI) strobe light (optional)
- Batteries

RECEIVER CPU BOARD

The receiver system electronics uses the same CPU board as the transmitter electronics. In the MITS mode, the STOM laser signals are decoded by the CPU board. If a valid hit is received, the strobe light flashes and the event is recorded in memory.

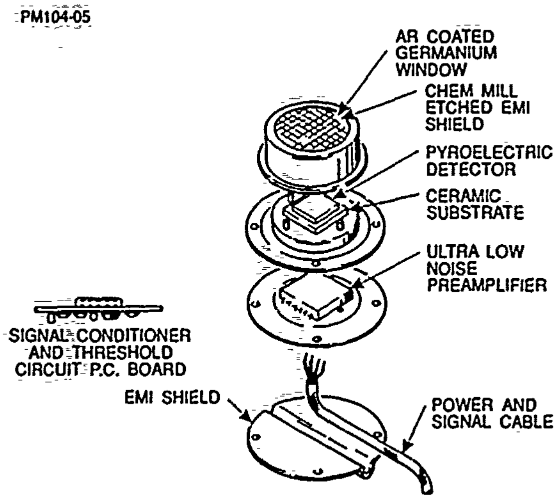


Figure 5. Schematic of Detector Assembly

In the M1/M1A1 mode, the CO_2 laser encoded message is converted into the MILES II code and injected into the standard MILES detector belts, which connect to the MILES II console. This code then appears to be a standard MILES transmission. STOM detectors are required on MILES equipment when the CO_2 laser is used because the MILES silicon detectors do not respond to the CO_2 laser radiation.

Figure 6. Module Assembly

STOM DETECTION SYSTEM ON M1/M1A1 TANK

A prototype of the M1/M1A1 system is being designed. Preliminary testing was done to determine the placement of the detector module and decoder box.

The design calls for two STOM detector modules to be placed on the front of the M1/M1A1 tank, two detectors on the back, and three detectors on each side. Connectors and cables are being designed for the M1/M1A1 STOM detection system.

The incoming STOM code is converted to a standard MILES II code, which is injected into the MILES II belt via an output connector on the decoder box.

TEST SET

A test set was designed for use in retrieving data from the laser controller assembly or the decoder box of the detection assembly. The functions performed in either unit are as follows:

- Setting the real-time clock to the time in the Epson Equity 386E laptop computer
- Clearing all data from the RAM
- Retrieving events for review
- Checking the status of the CPU board
- In field tests, recording laser transmission events that are time-tagged

The STOM test set consists of the following components:

- Epson Equity 286E laptop computer
- RS 232 cable
- "C" software
- IRIGB time decoder board (bc 630 AT)

LASER CODING

The pyroelectric detectors are energy detectors. During system analysis it was determined that a minimum of 7.5 mJ of laser energy was required to obtain ranges in excess of 6 km with the desired beam diameters. Since the present laser can transmit only 25 W peak power out of the optics, a 300 μs long laser pulse was used. This precludes the use of the standard MILES codes, since repetition rates of up to 8 kHz are required for the advanced MILES II codes.

A pulse position code (PPC) scheme was used. With this PPC there are two laser pulses per word. The first laser pulse designates the start pulse (P_s) and the second laser pulse designates the data pulse (P_d). Presently 10 words are sent per message. A message can be a tank round or part of an extended MISSILE tracking sequence. To obtain a valid hit, the detection of 3 words are required out of the ten sent. This helps reduce false alarms due to externally generated pulses.

Figure 7 shows a single STOM PPC word. The 300 μs long laser pulse limits the total number of possible weapons, ammunition types, and PIDs to 140.

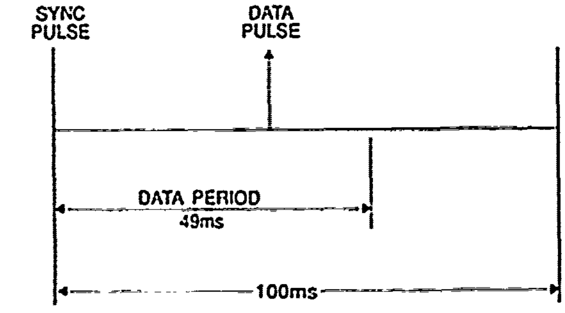


Figure 7. STOM Word

To overcome this defect and to permit more realistic training, a CO_2 "TEA" laser which is now being used in range finder applications is planned for the future. Existing "TEA" lasers have been tested for possible use with STOM. These lasers can deliver double pulses within a time period less than 8 ms. The measured pulse to pulse jitter can be kept below 2 μs, thus a 5 μs window in the PPC word can be used. This will permit over 8,000 distinct codes which can then be partitioned into various weapon, ammunition and PIDs. The use of the "TEA" laser can also permit the design of imbedded trainer transmitters, since the "TEA" laser rangefinder used in some existing weapon systems can also double as the MILES laser transmitter.

Both the present STOM rf laser transmitter and the "TEA" laser transmitters are eye safe.

SYSTEM TESTS

Various STOM system tests were performed during Phase II. These include the following:

- Temperature test
- Hit probability versus position test
- Obscurant tests

Temperature Test

Temperature testing on both the STOM receiver and transmitter was performed at temperatures ranging from -30°C to 62°C. The signal varied less than 10% from the mean value over the temperature extremes. There were no observed failures.

Hit Probability Versus Position Test

The final prototype equipment was used for hit probability versus position testing at the National Training Center (NTC), at Fort Irwin, California. The results (Figure 8) show the 95% hit probability diameter as a function of range. The computer generated profile is also shown.

PM104-04

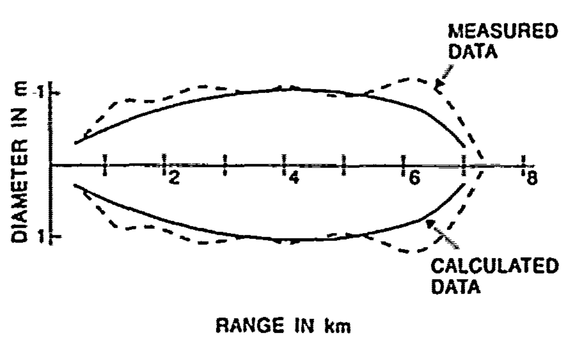


Figure 8. Hit Probability Diameter vs Range

This data was then used to predict the hit profile of a STOM laser transmitter firing at the front and sides of an M1/M1A1 tank equipped with two detector modules in the front, two in the back, and three on each side of the tank. These hit zones as a function of range are shown in Figures 9 and Figure 10.

PM104-02

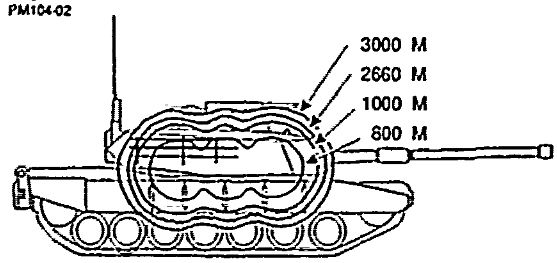


Figure 9. M1A1 Side Hit Profile

PM104-03

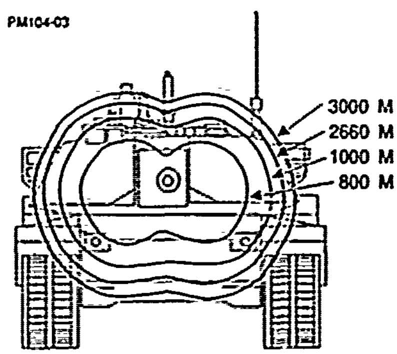


Figure 10. M1A1 Front Hit Profile

Obscurant Testing

During Smoke Week XIII at Eglin Air Force Base, the STOM system was tested in the presence of various battlefield obscurants. A standard Army M60A1 FLIR was used to view the target vehicle located at a distance of 1780 m. Video films were made of the scene as viewed from a standard day sight and from the FLIR. All data was time tagged with IRIGB data. The results of the testing verified that if you can see a target through obscurants with the FLIR, then you can obtain a hit with the STOM system during training exercises. A photograph generated from video film of the visually obscured target is shown in Figure 11. This same scene as viewed through a FLIR is shown in Figure 12.

SUMMARY

The major accomplishment of the STOM program was overcoming the necessity of using cooled detectors in the far infrared (10.6 μm) spectrum band. In addition to obscuration penetration, the hit profiles and maximum ranges obtainable are greatly improved over the present training systems, which use lasers operating in the near infrared. This results from the fact that near infrared laser transmitters have severe limitations on their power output because of eye safety constraints, whereas far infrared laser transmitters do not.

With the STOM system it is now possible to train on a battefield that has the same obscuration that would be found during a modern battle. This enables us to make the statement that if you can see it with the day sight, night sight, or FLIR, you can get a kill with the STOM system. This enables soldiers to truly train as they would fight.

Figure 11. Obscured View of Target

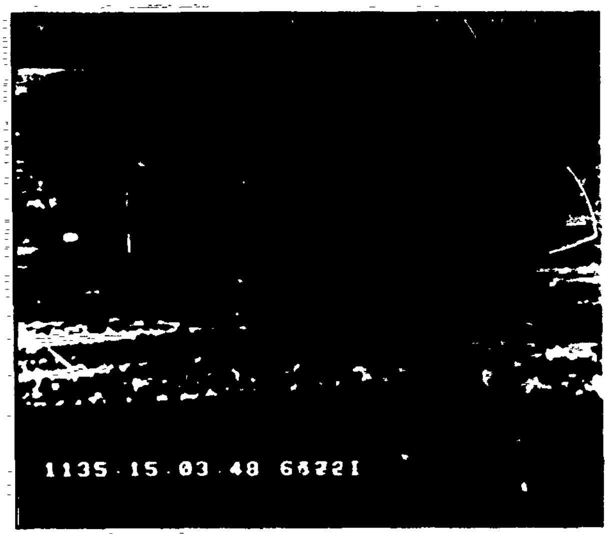


Figure 12. Unobscured FLIR View of Target

ACKNOWLEDGEMENT

The authors wish to thank Mike Sims for his assistance in taking and analyzing the obscuration data at Eglin Air Force Base.

BIOGRAPHY

James W. Surhigh, MS CALTECH, is Lead Systems Engineer for tactical engagement training at PM TRADE, Orlando, Florida. Mr. Surhigh has career experience in engineering, industry, and government as well as teaching experience on the secondary and college level in technical aspects.

Rudolph R. Gammarino is Chief Scientist for Training Devices for Loral Electro-Optical Systems (LEOS), Pomona, California. Previously, he was founder and president of Simulaser Corporation. He has over 29 years of experience conducting, managing, and directing numerous Army, Xerox, Simulaser, and Loral training and simulation programs, and he holds numerous patents for electro-optical systems and components. He was one of the primary developers of MILES and is the principal investigator for Loral's development of the STOM program. He has an MS in physics and has completed course work for a Ph.D. from New York University.

STANDARD PROTOCOL DATA UNITS FOR ENTITY INFORMATION AND INTERACTION IN A DISTRIBUTED INTERACTIVE SIMULATION

L. Bruce McDonald, Ph.D. • Christina Pinon Bouwens
Institute For Simulation and Training • University of Central Florida

Ronald Hofer, Ph.D. • Gene Wiehagen • Karen Danisas
Army Project Manager For Training Devices

Colonel James Shiflett
OSD Defense Models and Simulations Office

ABSTRACT

PM TRADE and DARPA have funded the University of Central Florida Institute for Simulation and Training (UCF/IST) to develop a Draft Standard for the Interoperability of Defense Simulations at the protocol data unit level. The second draft of the standard was completed in February 1991. The consensus of government and industry opinion was that the document represented a major step forward toward interoperability of dissimilar simulations. Based on inputs from government, industry and academia at four workshops, IST has developed a final draft standard for submittal as a DoD standard. This paper presents the contents of this standard, its intended use and its anticipated impact on the simulation and training industry. Discussion of the standard's contents include the protocol data units, their intended use and the underlying communications architecture. The paper also addresses future revisions of the standard and the attendant expanded capabilities.

INTRODUCTION

Over the last two decades, the United States military has developed an impressive array of simulation and training systems. These devices are extremely adept at training soldiers to do their jobs as individuals or as members of a small team. In addition, the test community has developed simulations that test the ability of equipment to perform its mission as an individual unit. However, the United States found in Grenada, Libya and Panama that the ability to perform a mission as an individual does not guarantee the ability to function as a member of a coordinated combined arms force.

The United States military has developed means for conducting large combined arms, multi-service exercises. However, these exercises are extremely expensive, subject to major environmental constraints, and can sometimes be interpreted as militarily provocative. DARPA and PM TRADE have developed Distributed Interactive Simulation (DIS), based on SIMNET technology, that will be used to train individuals to function as part of a large coordinated force. Also, DIS will allow the testing of developmental systems under realistic combat conditions. DIS can be used as a substitute for some field training and can allow for practice of warfighting skills when cost, safety, environmental and political constraints will not permit the field training required to maintain readiness.

DIS will take advantage of currently installed and future simulations manufactured by different organizations. Consequently, a means must be found for assuring interoperability between dissimilar simulations. The first step in achieving this interoperability is to develop a communications protocol. There must be an agreed-upon set of messages that communicate between host computers, the states of simulated and real entities, and their interactions.

DARPA and PM TRADE have funded the University of Central Florida Institute for Simulation and Training (UCF/IST) to develop a Draft Standard for the Interoperability of Defense Simulations at the protocol data unit level. The second draft of the standard was completed in February 1991. The consensus of government and industry opinion was that the document represented a major step forward toward interoperability of dissimilar Simulation.

A standard communications protocol must be developed for these dissimilar simulations to communicate with each other. The objective of the standard addressed in this paper is to define this communications protocol at the protocol data unit level. Since the emerging standard is primarily concerned with interoperability, the concept of Open Systems has become an

important issue. This subject has been dealt with quite thoroughly by the International Organization for Standardization (ISO), whose primary concern is the communications between heterogeneous computer systems developed by different vendors. ISO's efforts have led to the development of the Open Systems Interconnection (OSI) Reference Model. The standard was written with the assumption that the protocol will be implemented as part of the application layer of the OSI Reference Model.

The standardization process and recommendations for Distributed Interactive Simulation (DIS) are discussed below under the headings of History, Scope, Requirements, Protocol Data Units and Areas for Further Research.

HISTORY

The current work on standards began in August 1989 with the first workshop on Standards for the Interoperability of Defense Simulations. A second workshop took place in January 1990. As a result of these workshops and subsequent subgroup meetings, over 70 position papers containing recommendations for the standard were submitted to the Institute for Simulation and Training (IST). Using the work of SIMNET as a baseline and considering recommendations made in meetings and position papers, IST developed the first draft for a military standard which describes the form and types of messages to be exchanged between simulated entities in a Distributed Interactive Simulation (DIS). This draft standard was distributed to industry and government for review and comment in June 1990.

Workshop Reviews of Standard

A third workshop was conducted in August 1990 in which industry and government provided feedback on the proposed standard. These comments were incorporated into the standard and the final draft standard was submitted in January 1991 for approval by the workshop working groups. The working groups approved the final draft standard with minor changes, which are now being incorporated by IST. Recommended changes to the final standard are listed below.

Collision PDU - Change the mass field from 64-bit floating point to 32-bit floating point.

Update Threshold PDU - Delete this PDU until it is tested and include in later revision.

Radar PDU - Place in the optional section of the standard.

Detonation PDU - Articulated parts record should be included to indicate affected articulated parts. Remove energy, directionality and momentum fields.

Entity State PDU - Include force ID and guise type and modify articulated parts record.

MIL-STD Approval Process

This document will be submitted to the three military services to serve as the baseline standard. After approval by the services, the standard will be submitted for approval as a DoD standard. During this DoD standards approval process, the workshops will continue and a revision one and two will be developed with expanded capabilities. These revisions to the standard will also be submitted for approval as a DoD standard. We also intend to develop standards for the required correlation between simulated environments in different simulators as well as performance measures for evaluating the actions of the participants.

SCOPE OF STANDARD

The standard addressed in this paper establishes the requirements for data units exchanged between simulated elements in a distributed interactive simulation. It encompasses a portion of the application layer of a communications architecture as defined by the International Organization for Standardization's (ISO) Open Systems Interconnection (OSI) Reference Model.

REQUIREMENTS FOR DISTRIBUTED INTERACTIVE SIMULATION

The term Distributed Interactive Simulation refers to an architectural approach in which a simulation is distributed across a number of independent and self-sufficient computers instead of just one central computer. The term interactive reflects how these computers constantly interact by sending messages describing the current state of the simulation entities under their control. These messages allow the other computers to incorporate these state changes into their simulations.

Distributed Interactive Simulation can be defined as follows:

Distributed Interactive Simulation (DIS) is an exercise involving the interconnection of many simulation devices where the simulated entities are able to interact within a computer generated environment. The simulation devices may be present in one location, interconnected by a Local Area Network (LAN) or may be widely distributed on a Wide Area Network (WAN).

In order to fulfill its functional requirements, DIS must provide:

- Entity Information
- Entity Interaction

A brief description of each requirement follows:

Entity Information

Because of the great variety of simulated entities that will be involved in a single exercise, it is important to be able to transmit detailed information about each entity. This information

should include the entity's identity, its orientation, and its appearance to others. Below are classifications of types of information needed.

Types. Since a simulated entity can be a vehicle, a building, a missile, or even a cloud of smoke, a method for identifying the types of entities is needed.

Location and Orientation. The location, orientation, velocity, and acceleration (when appropriate) of a simulated entity are important for its representation by a computer. In order to keep network traffic within acceptable limits, the location and orientation information should contain velocity and sometimes acceleration. This information would allow the receiving computer to model (Dead Reckon) the position of the entity over time (based on last reported velocity and acceleration vector) without requiring constant updates over the network.

Appearances. The appearance of a simulated entity can be expressed in two ways: by the reflection of visible light or by the emission of acoustic or electromagnetic energy such as heat, radar, radio, etc. For example, besides its visual appearance, an entity can be identified by its unique infrared signature. If the exercise is taking place in the ocean, the entity can be identified by the sound it makes. Therefore, a method for communicating the different appearances of an entity is needed.

Entity Interaction

Throughout a simulation exercise simulated entities interact with each other. This interaction may take the form of weapons fire, update rate control, logistics support, or collisions.

Weapons Fire. When a simulated entity fires its weapon, its simulator needs to be able to communicate the location of the firing weapon and the type of munition fired. Depending on the munition type, the firing entity will determine the impact location. Given the munition type and the location of impact, all simulators can then assess their own entity damage.

Logistic Support. Other services such as resupply or repair of vehicles should be represented in a simulated exercise because of their significant impact on the outcome of military engagements. These functions and similar ones are provided by logistics support in a real battle scenario. Similarly, they should be provided by logistics support in a simulated battle.

Collisions. It is necessary to represent the collision of entities in a simulation. When a collision occurs, both entities need to be aware of the collision and each must determine any resulting damage to itself. DIS needs a way to communicate this type of collision information.

PROTOCOL DATA UNITS FOR DISTRIBUTED INTERACTIVE SIMULATION

DIS protocol is used by simulators to communicate information about the simulated world. Table I contains a list of the Protocol Data Units recommended for the standard. They are organized according to the information requirement category of which they are a part (e.g. entity information and entity interaction).

Table I
List of DIS Protocol Data Units

I. Entity Information

 A. Entity State PDU

II. Entity Interaction

 A. Weapons Fire
  1. Fire PDU
  2. Detonation PDU

 B. Logistics Support
  1. Service Request PDU
  2. Resupply Offer PDU
  3. Resupply Received PDU
  4. Resupply Cancel PDU
  5. Repair Complete PDU
  6. Repair Response PDU

 C. Collisions
  1. Collision PDU

A detailed discussion of these PDUs is beyond the scope of this paper. However, a brief discussion of a few important PDUs is presented. The two most important PDUs listed above are the Entity State PDU and the Fire PDU. Each of these example PDUs is discussed separately below.

Entity State PDU

A simulator periodically reports information about an entity it is simulating so that other simulators may correctly depict that entity. This information will be communicated using the ENTITY STATE PDU. Physical entities present in the simulation exercise include platforms, munitions, life forms, and environmental and cultural features. The various subcategories of entity types appear in Table II.

Table II
Entity Sub-types

Platforms Land Air Surface Subsurface Space **Munitions** Miscellaneous Detonator Ballistic Guided Anti-Air Anti-Armor Anti-Missile Anti-Radar Anti-Satellite Anti-Ship Anti-Submarine Battlefield Support Strategic Petroleum, Oil and Lubricants **Life Forms** SEALS Scouts Dismounted Infantry Categorized by Weapon Carried **Environmental** Smoke Fog Dust Flock of Birds Cloud Cloud With Rain Falling Cloud With Snow Falling Thermocline Knot School of Fish Whale School of Shrimp **Cultural Features** Bridge Building Defensive Embankment Crater Ditch

The Entity State PDU will be issued by a simulator when the following conditions exist:

1. The discrepancy between an entity's high fidelity model and its dead reckoned model exceeds a predetermined threshold (generally occurs when the platform changes its velocity vector).

2. A predetermined amount of time has elapsed since the issuing of the last PDU. The purpose of this issue is to inform new simulated entities of existing entities. It also serves to remind the existing entities that the issuing entity is still active.

Figure 1 lists field contents of the Entity State PDU.

FIELD SIZE (bits)	ENTITY STATE PDU FIELDS	
48	ENTITY ID	SITE - 16 - bit unsigned integer
		HOST - 16 - bit unsigned integer
		ENTITY - 16 - bit unsigned integer
8	PADDING	16 bits unused
8	FORCE ID	8 bits unsigned integer
64	ENTITY TYPE	ENTITY KIND - 8 - bit enumeration
		DOMAIN - 8 - bit enumeration
		COUNTRY - 16 - bit enumeration
		CATEGORY - 8 - bit enumeration
		SUBCATEGORY - 8 - bit enumeration
		SPECIFIC - 8 - bit enumeration
		EXTRA - 8 - bit enumeration
64	ALTERNATE ENTITY TYPE (GUISE)	ENTITY KIND - 8 - bit enumeration
		DOMAIN - 8 - bit enumeration
		COUNTRY - 16 - bit enumeration
		CATEGORY - 8 - bit enumeration
		SUBCATEGORY - 8 - bit enumeration
		SPECIFIC - 8 - bit enumeration
		EXTRA - 8 - bit enumeration
32	TIME STAMP	32 - bit unsigned integer
192	ENTITY LOCATION	X - Component - 64 - bit floating point
		Y - Component - 64 - bit floating point
		Z - Component - 64 - bit floating point
96	ENTITY LINEAR VELOCITY	X - Component - 32 - bit floating point
		Y - Component - 32 - bit floating point
		Z - Component- 32 - bit floating point
96	ENTITY ORIENTATION	Psi - 32 - bit BAM
		Theta - 32 - bit BAM
		Phi - 32 - bit BAM
256	DEAD RECKONING PARAMETERS	Linear Accel - 3 32 - bit floating points
		Angular Velocity - 3 32 - bit signal integers
		64 bits TBD

Figure 1 Entity State PDU

FIELD SIZE (bits)	ENTITY STATE PDU FIELDS (CONT'D)	
32	ENTITY APPEARANCE	32 - bits
96	ENTITY MARKING	Character set 11 Characters
32	CAPABILITIES	32 Boolean fields
24	PADDING	Unused
8	# ARTICULATED PARAMETERS	8 - bits unsigned integer
Varies	ARTICULATED PARAMETERS	Change - 16 bits
		ID - attached to - 16 bits
		Parameter type - 32 bits
		Parameter value - 64 bits

For each Articulated Parameter

Figure 1 Entity State PDU continued

The contents of each of these fields is described below:

General PDU Information - PDU Header

1. Protocol Version: Specifies the version of DIS protocol used in this PDU.

2. Exercise Identification: Specifies the Exercise to which the PDU pertains. This Feature allows multiple exercises to occur on the same network simultaneously.

3. Protocol Data Unit Type: Indicates the type of PDU to follow.

Static Entity Information

1. Entity Identification: Identifies the entity issuing the PDU.

2. Marking: Identifies any unique markings on an entity (e.g. a bumper number or country symbols).

3. Capabilities: Identifies the entity's capabilities in terms of logistics support to other entities.

Dynamic Entity Information

4. Time of Issue: Describes the time at which the PDU was issued.

5. Entity Appearance: Describes the dynamic changes to the entity's appearance such as on fire, destroyed, TOW launcher raised, etc.

6. Entity Location: Describes an entity's physical location in the simulated world.

7. Entity Velocity: Describes an entity's linear velocity in millimeters per second.

8. Entity Orientation: Describes the entity's orientation in terms of three angles.

9. Dead Reckoning Parameters: Elements of this field provide information required for dead reckoning an entity's position and orientation. These parameters consist of the following elements:

   Entity Acceleration. Describes an entity's linear acceleration in millimeters per second squared.
   Entity Angular Velocity: Describes the entity's angular velocity around its own axis.

10. Articulated Parts. Describes the orientation of each articulated part.

FIRE PDU

A FIRE PDU describes the type of munition fired, the location of the weapon from which it was fired, and the velocity of the munition. Also present in the PDU is the target range used for the fire control system and the kind of munition selected to aid analysis of the exercise. The contents of the Fire PDU are listed in Figure 2.

FIELD SIZE (bits)	FIRE PDU FIELDS	
16	EVENT ID	16 bit uns int
48	FIRING ENTITY ID	SITE ID - 16 bit uns int
		HOST - 16 bit uns int
		ENTITY - 16 bit uns int
48	TARGET ENTITY ID	SITE ID - 16 bit uns int
		HOST - 16 bit uns int
		ENTITY - 16 bit uns int
48	MUNITION ID	SITE ID - 16 bit uns int
		HOST - 16 bit uns int
		ENTITY - 16 bit uns int
96	BURST DESCRIPTOR	MUNITION - 32 bit uns int
		DETONATOR - 32 bit uns int
		QUANTITY - 16 bit uns int
		RATE - 16 bit uns int
96	LOCATION	X COORDINATE - 32 bit signed integer
		Y COORDINATE - 32 bit signed integer
		Z COORDINATE - 32 bit signed integer
96	VELOCITY VECTOR	X COORDINATE - 32 bit signed integer
		Y COORDINATE - 32 bit signed integer
		Z COORDINATE - 32 bit signed integer
32	RANGE	32 bit uns int

Figure 2 Fire PDU

The contents of the Fire PDU are described below:

1. Event Identification. Contains a number generated by the firing simulator to associate related events.

2. Firing Entity Identification. Identifies the firing entity.

3. Target Identification. Identifies the intended target.

4. Munition Identification. Gives an ENTITY ID to the munition. The Entity ID identifies the munition as a unique entity.

5. Burst Descriptor. Describes the type of munition fired, the quantity, and rate.

6. Location. Specifies the location from which the munition was launched.

7. Velocity Vector. Specifies the speed in millimeters per second and the direction of the fired munition.

8. Range. Specifies the range (in meters) that an entity's fire control system has assumed in computing the ballistic solution.

9. Timestamp. Specifies the time at which the data is valid.

Modeling the Trajectory of the Munition. There are no PDUs associated with the modeling of the trajectory of a munition. If the munition is the result of Direct or Indirect Fire, only its firing and detonation are reported. If the munition is a Guided munition, Entity State PDUs are transmitted for the munition throughout its flight.

Detonation of the Munitions. A Detonation PDU is issued when the trajectory of the fired munition is terminated. The firing simulator uses this PDU to inform other entities of the munition's detonation or impact location. In this way, other entities are FIGURE 3. Fire PDU informed of the munition's detonation so they might produce the appropriate visual and aural effects and assess damage.

Damage Determination. Once the location and type of detonation has been determined, each entity assesses its own damage based on its location in relation to the detonation. No PDUs are associated with this action.

AREAS FOR FURTHER RESEARCH

Communications Architecture

As stated earlier, the emerging Standard and the preceding workshops were primarily concerned with interoperability and Open Systems. The Open Systems Interconnection (OSI) Reference Model will be used throughout this section for discussion of communication architecture.

The interoperability requirements for communication have been well defined in the OSI reference model. However, DIS needs certain services not currently offered in available OSI protocols and so research must be done to develop them. These required services are described below.

1) Guaranteed Service for Real-Time Simulation. The requirements for DIS are based mainly on the needs of Real-Time Simulation, which requires information on a "timely" basis so that the representation and tracking of objects in the simulation can be accomplished as they are occurring. This requirement calls for a communication architecture that can deliver a message in a timely manner.

2) Multicasting Capabilities. In DIS it is sometimes necessary to send messages to a subset of nodes on the network. If a message is to be sent to all entities, it is sent using broadcast. If the message is to be sent to a specific group (as would be the case if more than one exercise taking place on the same network), the communications method used is termed multicasting.

These services are not currently provided in the OSI model.

3) Appropriate Security Levels. Security is an important requirement for DIS, but many problems remain unresolved. Some of these problems are related to how classified information may be securely transmitted. Others deal with how to keep the entire network secure. The current belief is that commercially available encryption software will be adequate for security requirements, but the real-time performance of this software may be too slow. Research on real-time performance of encryption software is required.

4) Connectionless Service. A connectionless service transmits data by simply sending the data out onto the network and addressing it to the entity(s) that require it. There is no need to establish a connection between simulation entities before transmitting data. This is a requirement for multicast service and is not currently provided in the OSI model.

Emitter PDU

The EMITTER PDU would be issued by the simulator for any platform possessing emitting capability. It is issued when the platform changes its velocity vector or changes the mode of one of its emitters. It is assumed that all simulators requiring emitter information have a database containing information about the operating parameters of emitters in each mode. An example database is the Universal Threat System for Simulators (UTSS).

When an EMITTER PDU is issued, it would include information about the state of all of its emitters for a particular database. Should an emitter from another database change modes, a separate EMITTER PDU would be issued.

Environment Information

For simulated entities to participate in the same exercise, they must all have access to the same environment information. Different types of information about the environment are necessary to make the exercise as realistic as possible. This information may include changes in the terrain, weather, and ambient illumination.

Changes in the Terrain. During the course of a real battle, changes in the terrain occur frequently. An explosion may create a crater or blow up a bridge. Ditches might be dug and defensive embankments may be built. In addition, cultural features such as bridges and buildings could be destroyed or built. All of this information must be available to the participants in a simulated battle just as it would be accessible in a real battle. Therefore, DIS must provide the necessary functions to support dynamic terrain.

Weather Conditions. Weather conditions affect real life battle scenarios. Similarly, they should have an effect on the simulated battle. Conditions such as wind, rain, snow, fog or clouds need representation in a simulated exercise. The wind and its effect on a cloud of smoke that affects visibility or chemicals that affect dismounted infantry need to be represented as well.

Proposed Simulation Management Protocol

IST proposes a Simulation Management Protocol (SIMAN) that could provide many of the services required by DIS.

SIMAN would perform the following functions:

1. Exercise setup
2. Exercise start/restart
3. Exercise maintenance
4. Exercise end

SIMAN would serve as a central database for the simulation. It would record the exercise for purposes of playback, restart and admittance of new entities to the exercise. SIMAN would also perform data logging functions such as updating its database on entity capabilities and changes in the terrain. Further research is required into the requirements for SIMAN functions and the most efficient means of providing these services.

Unmanned Forces

One type of entity that is represented in a simulated battle is the Unmanned Force or Semi-Automated Forces (SAFOR). As simulated entities in the exercise, unmanned forces have many of the same requirements as manned forces. The data messages (PDUs) communicated on the network are the same as those for manned simulators. Unmanned forces, however, have some unique informational and database requirements that other entities do not have. Further research is required before effective semi-automated forces can be added to DIS.

Issues Concerning Fidelity Measures

Fidelity Measures address the allowable delay between operator action and simulated response, as well as the required fidelity for representing the visual appearance or sensor imagery of an entity or the environment. Many fidelity measures issues have been resolved in previous research on individual operator training systems. The three most critical remaining DIS fidelity issues requiring research are delay, entity appearance at long ranges, and depiction of environmental appearances.

Delay. The allowable delay between operator action and simulation response will depend on the criticality of the task being executed by the operator. One of the most time-critical tasks in distributed interactive simulation is tracking a target just prior to firing a weapon. Consequently, the smallest acceptable delay in a DIS will be that between the issuance of an appearance PDU by a target entity and the display of that entity's location on the engaging entity's display. Determination of acceptable delay will require empirical studies of operator performance under varying delay conditions.

Entity Appearance At Long Ranges. One shortcoming of current distributed interactive simulation is that the displays have insufficient resolution to accurately depict entities at long range, thereby preventing the engagement of these entities at a range specified in doctrine. This problem may be solved by using higher resolution displays or by color coding images too small to identify. Determining acceptable means of increasing target identification ranges will require empirical studies of operator performance with alternative modifications to the current approach.

Depiction of Environmental Appearance. The appearances of environmental entities such as smoke, fog, clouds, rain and snow need to be depicted in a manner realistic enough to achieve the training or equipment evaluation objectives. Each of these environmental entities effects visibility to a varying degree based on the density of the entity.

Five levels of density should be sufficient to meet the training and equipment evaluation objectives. Definition of how the visual system will depict the density of these environmental entities should be based on target detection range for each level of density. For example, "Fog with density level three shall produce a 50% target detection probability for the T-72 tank at ____ meters." Further research is required before these values can be stated.

Update Rate Control. The frequency at which one simulated platform must transmit an update of its location and orientation or its emitter status to another platform depends on what task the operator of the simulator is attempting to execute. If the operator of one platform is simply observing the other platform's motion for identification, the exact location of the platform is less critical and frequent updates are not required. However, if the operator is tracking the other platform in preparation for firing or two platforms are flying in close formation, the exact location is critical and a higher update rate is required. DIS needs a means of controlling platform location and orientation update rate in order to meet the requirements for some critical operator tasks without overloading the network while the operator is executing less critical tasks.

CONCLUSIONS

With the increased operating costs of military equipment and reduced budgets, increased use of simulation is needed to maintain readiness. Distributed Interactive Simulation will allow the armed forces to use the installed base of individual training devices to perform large scale team training exercises in a manner similar to SIMNET. The Simulation PDUs of the SIMNET protocol were considered as a baseline for this effort. The University of Central Florida's Institute for Simulation and Training has prepared a Standard (at the Protocol Data Unit level) which will allow dissimilar simulations to interoperate in a Distributed Interactive Simulation.

ABOUT THE AUTHORS

Dr. Bruce McDonald is the Program Manager on the Interoperability Standards program. He has a Ph.D. in Industrial Engineering and 23 years of experience in research and system analysis on training systems and operational equipment. For the last 14 years, he has concentrated on training systems and has completed the following tasks: functional descriptions and detailed specifications for training systems, requirements analysis of war game trainers and instructor station designs. He was program manager for a Catapult Launch Systems Trainer (Device 11F12) and the Principal Investigator on a study to develop Embedded Training capabilities for the AN/SPA-25G Radar system.

Christina Pinon Bouwens is the Program Engineer on the Standards program. She has an M.S. in Mathematics Science and has seven years of experience. She is currently working on communications protocols to support Distributed Interactive Simulation.

Dr. Ronald Hofer is the Chief Scientist at the Army's Project Manager for Training Devices. Dr. Hofer has more than 25 years' experience in simulation and training; he has spent 15 years in this field at PM TRADE. He is also chairman of the Communications Protocols working group which developed recommendations for this standard.

Gene Wiehagen is a System Engineer at the Army's Project Manager for Training Devices and is the BDS-D ATTD Manager in charge of all networking related projects. Mr. Wiehagen has a B.S. in Engineering aand is a member of the Steering Committee for the interoperability of defense simulations.

Karen Danisas has a B.S. in Physics, an M.S. in Simulation Engineering, and has six years of experience in the simulation industry. She is an engineer at the Army's Project Manager for Training Devices and is in charge of standards development for DIS.

Colonel James Shiflett is the Technical Director of the Defense Models and Simulations Office in the Office of The Secretary of Defense and was previously a program manager at the Defense Advanced Research Projects Agency where he was responsible for distributed interactive simulation technology development.

THE CAPABILITY OF THE DISTRIBUTED INTERACTIVE SIMULATION NETWORKING STANDARD TO SUPPORT HIGH FIDELITY AIRCRAFT SIMULATION

Edward P. Harvey
BMH Associates, Inc.
Norfolk, VA

Richard L. Schaffer
BBN Systems and Technologies
Cambridge, MA

ABSTRACT

How will U.S. tactical aviation forces train for future conflict? The prevailing budgetary climate will force a reduction in the frequency of training operations using actual equipment for some time to come. One cost-effective means for U.S. combat forces to conduct training is through the application of distributed simulation technology. A large scale simulation network which is based on the new Distributed Interactive Simulation (DIS) draft military standard for simulator networking and is accessible by the components of all three services will be the likely medium for conduct of this type training.

DIS networking protocols evolved from ground vehicle networking protocols developed during the U.S. Army/ DARPA SIMNET program. It is therefore understandable that some misconceptions may exist over the capability of DIS to provide sufficiently accurate vehicle position and orientation data for high performance aircraft simulation. High performance tactical aircraft simulation requires a high degree of vehicle position and orientation accuracy for conduct of fully effective training. Operational community acceptance is dependent upon the capability of a DIS network to support all potential high performance aircraft combat interactions including air-to-air missile engagements and air-to-air gunnery.

This paper will quantitatively detail DIS vehicle position and orientation accuracies throughout the potential range of simulated aircraft maneuvering capability. Entity State (position/orientation) Protocol Data Unit (PDU) transmission frequencies for differing order Dead Reckoning (DR) algorithms will be empirically derived for the F-16 fighter aircraft performing the dynamic Paris airshow flight routine. Average Entity State PDU transmission frequencies will be presented as a function of dead reckoning algorithm threshold values. This data will show the capability of the DIS networking standard to support high fidelity aviation training tasks, even those requiring precise real-time position updates such as air-to-air gunnery, while achieving significant network bandwidth reductions.

INTRODUCTION

You've just come from the Intell shop and there is no news on NAIL 21, the F-16 which was lost on the strike mission earlier in the day. Brief, launch, transit to the tanker, and ingress were no problem. It sure helped to have had 7 missions under your belt. The first few were a real zoo. Forming up a strike package from a half dozen different fields including the Navy carrier airwing assets, was something you didn't practice for during peacetime training. The morning mission had gone smoothly until the bandit warning from COUGAR, the AWACS controlling the strike. COUGAR had vectored your division of F-15C aircraft to intercept a flight of four bad guys and each of your wingmen splashed a single with AIM-7 shots in the face. Your Sparrow appeared to guide, but must have failed to fuse, and your MiG was the only one of the four to escape. You wanted to chase him down so bad you could hardly stand it but the mission comes first so you broke off the pursuit, reformed your division, and got a vector from COUGAR to rejoin the strike package.

Unfortunately, the target area was hot. Plenty of AAA, but that was going off well below everyone's altitude so it wasn't a big concern. Then at the roll-in point there was a flurry of SAMs launched nearly simultaneously and one managed to guide on NAIL 21. Either NAIL 21's EW pod was on the fritz, or maybe it was just a lucky ballistic shot. Nobody had any RHAW indications. At any rate there was a man in the chute over 200 miles deep in bad guy territory. Hopefully, that wouldn't happen this afternoon.

Back to the present. Time to brief your division for the afternoon go. This time you wouldn't be tied to the strike package and were going on a sweep. Hopefully, everything would go as well as this morning, except for the loss of the F-16 and not getting that kill. You've got to do this right. The boss is watching to see who are the performers. You stop for a moment's reflection. It is a weird feeling that this war is taking place in simulation facilities at twelve different USAF and USN bases in CONUS and Germany. Who would have ever thought....

DISTRIBUTED INTERACTIVE SIMULATION

The scenario recounted in the previous paragraphs is an example of the type of realistic team training which will be possible upon implementation of the Distributed Interactive Simulation (DIS) standard for networking simulators. The DIS standard for networking simulators, now released in draft form, is a direct outgrowth of the simulator networking protocol developed during the

DARPA/U.S Army SIMNET program. The University of Central Florida Institute for Simulation and Training (UCF IST) is the organization responsible for finalizing the content of the DIS standard for release to industry. UCF IST organized a series of semi-annual industry working groups to maximize participation in the DIS standard development process. These DIS working groups first convened in August 1989 to make recommendations for enhancing the existing SIMNET networking protocol to accommodate all classes of simulated vehicles and systems for network team training applications.

Over 70 point papers were submitted for review after the first DIS working group session. Many of these papers contained recommendations for alternate networking approaches; some addressed perceived weaknesses in the SIMNET protocols; and other papers advocated development of additional networking protocols. Many lively discussions by members of industry ensued which resulted in the submission of additional point papers to defend or attack positions taken, or recommend enhancements to the standard. As the DIS standard matured fewer point papers were submitted for each subsequent DIS working group session. Only 8 point papers were submitted for the working group session held in March 1991 with the majority of these papers addressing refinements to the existing draft DIS standard.

The DIS standard development process has been a success. A simulator networking standard has finally been developed by industry. It is interesting to note that after nearly two years of effort, the DIS standard is surprisingly similar to the original SIMNET protocols from which it evolved. This is remarkable since SIMNET networking protocols were developed primarily for an armored vehicle simulator network while the DIS protocols must support all classes of simulated vehicles, including high performance aircraft simulators.

The next section of this paper describes the DIS Entity State PDU and the role of dead reckoning. This is followed by a discussion on network protocol design considerations and a comparison of the performance characteristics of armored vehicles and high performance aircraft which influence network protocol design.

DIS Entity State PDU And Dead Reckoning

DIS simulators exchange messages with each other using a communications network. The message used to communicate vehicle state information is called the Entity State PDU. The DIS Entity State PDU is an enhanced version of the SIMNET Appearance PDU. The Entity State PDU contains all information needed to depict the originating vehicle. This information includes the type of vehicle and its position and orientation. In addition, it contains a set of parameters, called dead reckoning parameters, that are used to extrapolate the position and orientation of the vehicle into the future. This model of the future position of the vehicle is known as the dead reckoning model. When this model is computed in a remote simulator, it is referred to as the remote vehicle approximation (RVA). The procedure by which the dead reckoning model is calculated is known as the dead reckoning algorithm. The inputs to the dead reckoning algorithm are the position, orientation and dead reckoning parameters from the most recent Entity State PDU and the current time. DIS supports more than one kind of dead reckoning algorithm.

The Entity State PDU is not transmitted at the frame rate of the originating simulator. Instead, a simulator transmits an Entity State PDU only when the discrepancy between its own high fidelity model of its position and orientation and that of the dead reckoning model exceeds a certain threshold. In DIS, there are six separate thresholds; a positional threshold along each body axis, and a rotational threshold about each body axis. If any of these thresholds are exceeded, a new Entity State PDU is transmitted to the other simulators on the network.

Benefits of Dead Reckoning. The use of dead reckoning for networked simulation was pioneered by SIMNET. The principal motivation for the use of dead reckoning is to reduce the network bandwidth required to support a given application. From another viewpoint, dead reckoning increases the scale of exercise that can be supported on a given network. In the case of SIMNET ground vehicles, the reduction in network bandwidth requirements is dramatic, approximately 83% of network traffic is eliminated through the use of dead reckoning [Miller, et. al. 1988]. Another benefit of dead reckoning accrues to receiving simulators. In general, there is a computational cost associated with receiving and processing a PDU. By substantially reducing the rate at which PDUs are received, dead reckoning likewise substantially reduces this computational cost.

Finally, the dead reckoning model explicitly defines the state of remote vehicles in the intervals between the reception of Entity State PDUs. This provides an unambiguous definition of the state of remote vehicles, independent of the frame rate of the underlying simulators. Thus, it provides a means for simulators operating at different, and even irregular, frame rates to interact.

Tradeoffs Among Computation, Fidelity, and Network Bandwidth. Dead reckoning allows an explicit tradeoff to be made among three factors; network bandwidth requirements, computation performed in the RVA, and the fidelity of remote vehicle position and orientation. Network bandwidth is consumed by the required rate of transmission of Entity State PDUs. The cost of computation is the cost of computing the dead reckoning model for all remote vehicles within a range of interest. There is also a cost to encode the dead reckoning parameters on the part of the Entity State PDU sender. However, this cost is generally negligible since it need be done only once per Entity State PDU transmission, rather than once per frame per remote vehicle. Finally, the fidelity of representation of remote vehicles is simply determined by the dead reckoning thresholds for position and orientation.

There are two parameters which control these tradeoffs. One parameter is the choice of dead reckoning thresholds. These thresholds provide control of the tradeoff between network bandwidth and the fidelity of remote vehicle position and orientation representation. As the thresholds are decreased, the discrepancy between the

dead reckoning model and the internal high fidelity model more quickly reaches its limit, the threshold, consequently, the rate at which Entity State PDUs are transmitted increases.

The other parameter controlling the tradeoffs is the choice of dead reckoning algorithm. This choice determines the tradeoff between network bandwidth and computational cost. More complex dead reckoning algorithms can better model the future path of the vehicle. Consequently, the discrepancy between the internal and dead reckoning models accumulates less quickly. The resulting reduced Entity State PDU transmission rate is bought at the cost of increased computation required by the more complex dead reckoning algorithm. This cost may be significant, since it must be computed each frame for each remote vehicle.

For any system design, and for any particular exercise, these two parameters may be adjusted to maximize fidelity within the limits set by the computational and network resources available.

Simulation Network Protocol Design Considerations

When SIMNET was developed it was postulated that networking armored vehicle simulators would be relatively simple compared to networking high performance aircraft simulators. This seems logical at first thought; after all, tanks are certainly much slower and appear to be less maneuverable than high performance aircraft. Therefore, an armored vehicle simulator network should require a lower entity state update rate than a high performance aircraft simulator network to provide sufficient fidelity for conduct of effective training. Based on a cursory inspection of the problem, it is natural to assume a network communications protocol developed for an armored vehicle simulator network would not have sufficient performance for conduct of effective training in a high performance aircraft simulation network. In fact, this is not the case for reasons which are discussed in the following paragraphs.

Vehicle Operating Characteristics. Each simulated entity must be able to detect the actions of other entities on the network in real time in order to provide adequate training fidelity This may be achieved by the brute force method of transmitting entity state data at the frame rate of the transmitting simulator. In this manner the maximum delay which may be encountered before an entity action will be transmitted on the network is the frame period of the transmitting simulator. This method of ensuring real time entity state data transmission has its limitations for very large simulator networks. Affordable computational resources and available network bandwidth eliminate this method as a viable approach for a simulator network with many vehicles. A more clever approach than a high entity state data transmission frequency is necessary to provide real time entity state updates over a large scale simulator network. Below, we investigate whether one can take advantage of the characteristics of the various types of simulated vehicles which may be networked.

Vehicle Speed. Armored vehicles operate in a much slower speed regime than high performance aircraft by an order of magnitude or more. Armored vehicles also appear to be less maneuverable than high performance aircraft. It may seem that vehicle speed is the factor upon which the entity state transmission frequency of a simulator should be primarily dependent. If vehicle speed is a primary factor, high performance aircraft simulators would need to transmit entity state data at a much higher rate than armored vehicle simulators. Since aircraft are at least an order of magnitude faster than armored vehicles, the aircraft simulator entity state data transmission frequency should be at least ten times that of on armored vehicle.

This assumption is wrong. For example, a simulator using a simple dead reckoning algorithm can accurately calculate the future position of a simulated vehicle in steady state motion with a constant velocity regardless of the speed of the vehicle. As long as a simulated high speed aircraft remains in steady state flight there is no need for a high entity state data transmission rate to allow other simulated vehicles on the network to accurately calculate the future position of this simulated aircraft. Vehicle speed is not the dominant factor upon which to base design decisions regarding the entity state data transmission frequency of a simulator network.

Vehicle Maneuverability. Although an armored vehicle is maneuverable in two dimensions while an aircraft is maneuverable in three dimensions, there are several similarities between the behavior of the two types of vehicles. The maximum rate at which an armored vehicle can change its orientation through rotation about its vertical axis is similar to the maximum rate an aircraft can change its orientation by rolling about its longitudinal axis. An armored vehicle can only change its orientation at the maximum rate if it is rotating about a point in a constant position. An aircraft performing a maximum rate rolling maneuver about its longitudinal axis does not alter its flight path or future position appreciably. Although each type of vehicle is changing its orientation as rapidly as is possible, the future position of the vehicle is not appreciably affected, the tank remains in the same position and the aircraft direction of flight remains virtually constant.

In addition to the similarity in the rate at which the two types of vehicles can change their orientation, there is a similarity in the rate they are both able to change direction (turn) while in motion. A moving tank and an aircraft in flight are limited in their ability to change their direction of motion as a function of speed. The faster the tank and the aircraft travel within their respective speed ranges, the slower the possible turn rate. High performance aircraft are typically G-limited and rarely exceed turns which generate 9.0 Gs. An F-16 aircraft can generate a 9.0 G turn, which equates to a turn rate of approximately 22°/sec, at a speed of 440 knots. As aircraft speed increases the turn rate of the aircraft decreases at maximum G. At 600 knots an F-16 at 9.0 Gs turns at a rate of 16°/sec, a 27% decrease in turn rate from the possible turn rate at 440 knots airspeed. Tanks are able to generate similar turn rates as high performance aircraft as they are traveling much slower and do not need to generate much radial G to achieve equivalent turn rates. A tank traveling at 20 miles per hour need only generate 1/3 G to achieve a turn rate of 22°/sec.

There is much similarity between the rates at which armored vehicles and high performance aircraft can change their orientations and direction of travel. A networking protocol which takes advantage of the similar manner in which these two different classes of simulated vehicles change orientation and direction of motion should be capable of providing sufficiently accurate remote vehicle position and orientation updates over the network.

Tactical Considerations for Armored Vehicle Simulators

An armored vehicle crew increases their exposure to danger from adversary infantry, armored vehicles, and aircraft whenever they move their vehicle. Therefore, armored vehicles spend much of the time in prepared firing positions hidden from these foes. Entity state data which accurately describes a stationary armored vehicle's position and appearance can be transmitted at a relatively low frequency over a simulator network. If a vehicle remains stationary and does not change its appearance in any way to an outside observer it only need transmit one entity state information package to the rest of the network at the moment the vehicle assumes this state to provide "ground truth". In practice, the DIS and SIMNET networking protocols require stationary vehicles to transmit an entity state update at least every 5 seconds so that simulated vehicles joining the network must wait only 5 seconds to receive all remote vehicle data necessary to calculate "ground truth".

The motion of an armored vehicle is affected by not only the driver of the vehicle, but also by the variations in the terrain over which the vehicle moves. Armored vehicles also have dynamically moving articulated parts including turrets and cannons which may rotate and elevate at a rate greater than the rate which the vehicle itself can change its orientation. The orientation of articulated parts must also be communicated which further increases the entity state data transmission rate for armored vehicles.

Tactical Considerations for High Performance Aircraft Simulators

A manned high performance aircraft is constantly in motion and fully capable of maneuvering, other than the brief time periods spent taxiing before and after the mission. Although an aircraft is in continuous motion while in flight, the motion is predictable for the majority of the time. The flight path of an aircraft is only unpredictable during the time interval in which a pilot repositions the flight controls to perform a maneuver. Once the aircraft enters a steady state maneuver the flight path again becomes predictable. For example, a pilot performs three flight control movements to initiate a steady state high G turn; a roll input to start the turn entry, a pitch input to generate the G, and a roll input to maintain a steady turn attitude while holding the pitch input to maintain G. During an ideal 9.0 G, 360° horizontal turn, an F-16 aircraft is in predictable, steady state flight for all but a few seconds. approximately 2 seconds to enter the turn and 1 second to return to wings level flight after the turn is complete. (It takes slightly more time to enter the turn as the G onset rate is limited by the flight control computer.)

A typical fighter aircraft mission profile includes 2 to 4 hours of transit time to and from the target area with perhaps 4 minutes of air combat maneuvering (ACM). ACM is the flight regime which was once assumed to be impossible to support using the DIS networking protocol ACM is actually a series of predictable, steady state flight maneuvers such as the high G turn described in the previous paragraph. The aircraft is in unpredictable flight only for a very small percentage of the time. (In this context the term "predictable" is used to describe the flight path of the simulated high performance aircraft for a long period of time in the future for simulation purposes, perhaps 1 - 2 seconds.) The only unpredictable force which can affect the flight path of the aircraft is the pilot and he/she generally points the aircraft and then lets it fly steady state over the small periods of time we are concerned with in simulation.

Protocol Considerations Summary

Other than speed, which is not the dominant factor in protocol design, armored vehicles and high performance aircraft have similar performance characteristics including similar capabilities to change vehicle orientation and direction of travel. High performance aircraft remain predictable for the majority of the time due to the steady state nature of aircraft flight. Vehicle motion for armored vehicles is not quite as predictable as for high performance aircraft as variations in the terrain over which the vehicle travels affect the orientation of the vehicle and armored vehicles typically have tactically relevant articulated parts which must be accurately represented to the rest of the world.

It appears "predictability" is the key to development of a robust simulator networking protocol. Both armored vehicles and high performance aircraft behave predictably for the majority of the time. The position and orientation of "predictable" vehicles can be determined well into the future, several seconds at least, with sufficient accuracy for conduct of fully effective training. Network bandwidth can be efficiently utilized by transmitting entity state data only when there is a change in the state of a vehicle. All other vehicles on the network assume the transmitting vehicle will remain in steady state motion until they receive the next entity state update from that vehicle. Use of dead reckoning to locally calculate the position and orientation of remote vehicles at the local host frame rate should provide a substantial reduction in the amount of entity state data transmitted over the network while providing high fidelity remote vehicle position and orientation accuracy. Design of a dead reckoning algorithm which provides sufficient vehicle position and orientation accuracy for any training simulation network is possible.

As will be seen from the results of the experiments detailed in the following sections of this paper, the DIS networking protocol is a very robust networking protocol which is capable of providing sufficiently accurate vehicle position and orientation data at a sufficient rate over local and long haul networks to support high performance aircraft simulation.

PROTOCOL DATA UNIT TRANSMISSION FREQUENCY EXPERIMENT

The experiment was based on flight dynamics data from a General Dynamics F-16 flight simulator. Time-averaged Entity State PDU transmission rates were measured over a range of thresholds for three different dead reckoning algorithms. These PDU transmission rates document the degree to which dead reckoning can reduce network bandwidth requirements even when relative tight thresholds are employed.

In the late 1970's General Dynamics demonstrated the high performance maneuvering capabilities of the F-16A Falcon fighter aircraft at the Farnborough and Paris Airshows. This airshow flight routine consisted of a series of maneuvers very similar to those performed during classic maneuvering air combat engagements. Some of these maneuvers, in particular the high-G turns, approached the maneuvering limits for a piloted aircraft. Even now, the F-16 is still one of the highest performance fighter aircraft currently flying; therefore, an F-16 flight dynamics simulation should provide the "worst case" for Entity State PDU transmission frequency for a given order dead reckoning algorithm within the high performance aircraft class of networked simulators.

Maneuvering data for this experiment was collected by General Dynamics during a five minute duration F-16 simulator session flown by a fighter pilot with over 1,500 hours of experience in the F-4 and F-16 fighter aircraft. The GD F-16 flight simulator consisted of two Harris 1000™ and three Harris 800™ host processors, and a FPS 5000™ aero model. The simulator visual system was an Evans and Sutherland CT-6™ in a 24 foot dome. The frame rate of the F-16 simulator and the data collected is 50 Hertz.

Figure 1 depicts the actual flight routine performed for this experiment. The pilot conducted a maximum performance take-off transitioning immediately into a slow speed loop. After performing the slow speed loop the pilot performed a series of level rolls then pulled up into a vertical climb followed by a vertical reversal and acceleration in level flight into a high performance level turn. The pilot then reversed heading twice out of the high performance turn and accelerated down the runway in the opposite direction from takeoff. At the departure end of the runway the pilot performed a pull-up and a series of vertical rolls to slow the aircraft before landing.

The flight path data was analyzed by a program which, for a given dead reckoning algorithm, counts how many Entity State PDUs would be transmitted over the duration of the airshow flight routine. For each frame of data, the current position and orientation of the aircraft were compared to the dead reckoning model as applied to the last Entity State PDU transmitted. If the position or orientation exceeded the selected thresholds, a new Entity State PDU was counted as being transmitted. For each dead reckoning algorithm, the position thresholds were varied from 3 meters to 3 centimeters and the orientation thresholds were varied from 10 degrees to 0.1 degrees.

Packet rates were calculated for three different dead reckoning algorithms. These dead reckoning algorithms incorporated either first or second order dead reckoning of position and either zeroth or first order dead reckoning of orientation. "Zeroth order" dead reckoning of orientations simply means orientation was maintained constant over the frame interval. First order dead reckoning of orientation consists of integrating angular velocity over the frame interval to update orientation. Similarly, first order dead reckoning of position consists of simply integrating linear velocity over the frame interval. Second order dead

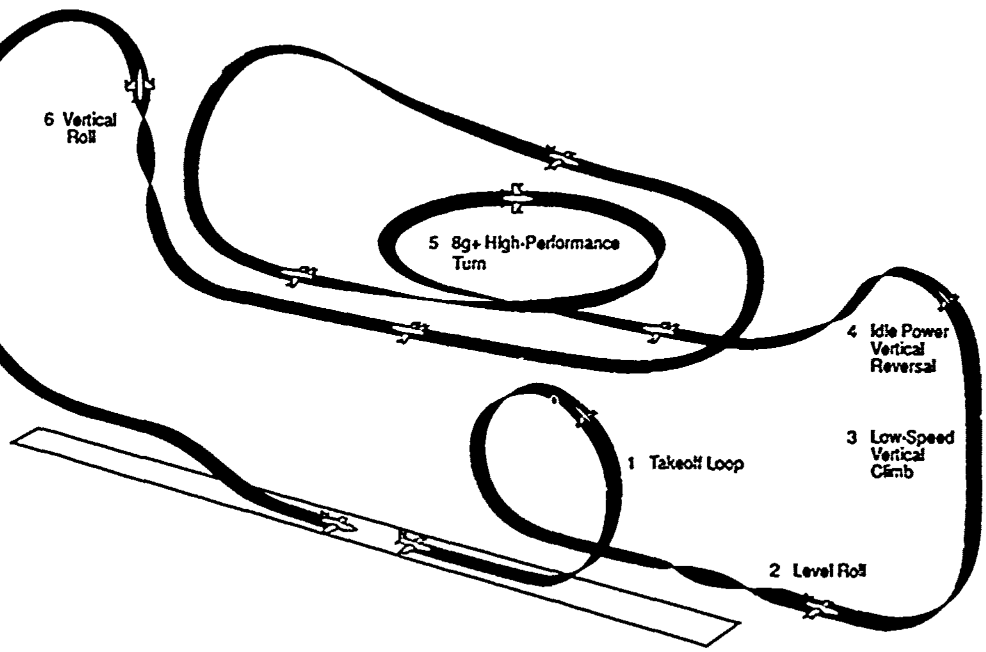


Figure 1: F-16 Paris Airshow Flight Routine

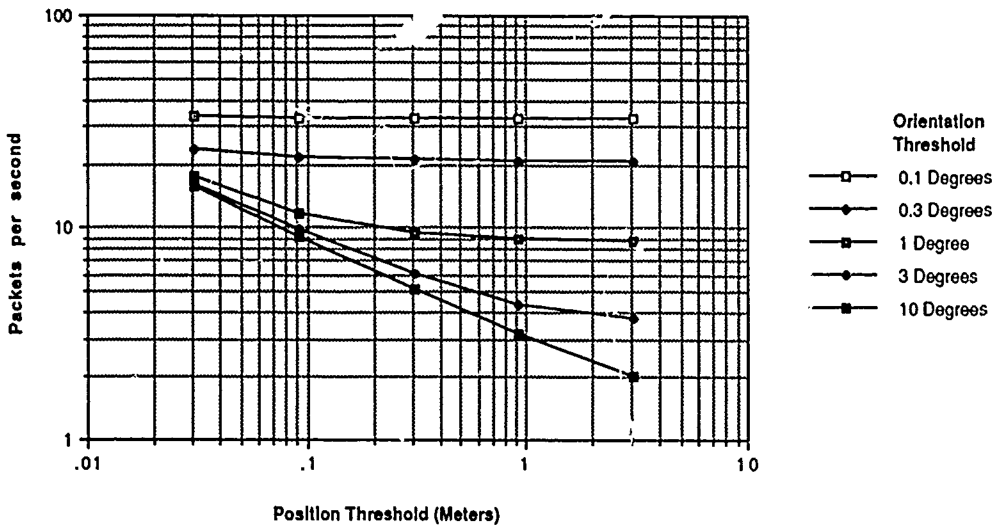


Figure 2: Algorithm 1: First Order Position, Zero Order Orientation

reckoning of position consists of integrating acceleration over the frame interval to update velocity and then integrating velocity to yield position. The average Entity State PDU transmission rate over the course of the flight routine is plotted for the range of thresholds for differing order dead reckoning algorithms in Figures 2 through 4.

Experiment Results

Figure 2 is a plot of the Entity State PDU transmission frequency versus vehicle position thresholds using a first order position, zero order orientation dead reckoning algorithm for varying vehicle position and orientation thresholds This dead reckoning algorithm is the same one used in SIMNET. The position thresholds were equal for the vehicle x, y, and z axes, and the orientation thresholds were equal for roll, pitch, and yaw. The PDU transmission frequency shown is the average encountered for the complete five minute F-16 Paris Airshow flight profile.

In Figure 2 it can be seen that an average Entity State PDU transmission frequency of approximately 4 packets per second is achieved using the original SIMNET armored vehicle dead reckoning algorithm with a vehicle position accuracy of 1 meter and an orientation accuracy of 3 degrees. It is important to note that the dead reckoned position of the aircraft remains within an 8 cubic meter volume, with the actual aircraft position being the centroid of the volume, for the entire flight profile with only a 4 packet per second PDU transmission frequency. Increasing vehicle position accuracy by an order of magnitude, from 1 meter to .1 meter, only doubles the Entity State PDU transmission frequency for a 3 degree orientation threshold.

The highest accuracy data derived, .03 meter position and 1 degree orientation, equates to an Entity State PDU transmission frequency of approximately 17 packets per second. This accuracy is likely sufficient for even engineering simulation and far exceeds the accuracy required for simulation training exercises. The use of a first order dead reckoning algorithm identical to the SIMNET armored vehicle dead reckoning algorithm for this F-16 flight profile provides a 66% (approximately) reduction in network traffic from 50 Hertz to 17 Hertz even using extremely very tight thresholds of .03 meter for position and 1 degree for orientation.

Figure 3 is a plot of the Entity State PDU transmission frequency versus vehicle position thresholds using a first order position, first order orientation dead reckoning algorithm. The most striking feature of the plot is the significant reduction in PDU rate for tight angular thresholds. Adding first order extrapolation of orientation is so successful that violations of the position threshold are the primary source of PDUs, except for position thresholds looser than about a meter. Thus, the next natural step to reduce PDU rates is to improve the predictive power of the algorithm with respect to position.

Figure 4 is a plot using a second order position, first-order orientation dead reckoning algorithm. Note that the scale on the vertical axis has been reduced by a decade. Packet rates due to positional threshold violations have been dramatically reduced. For example, the curve connecting points of 0 1 degree orientation threshold is now nearly flat While the angular-threshold-dominated right hand end of this curve has changed very little, the packet rate at the position-threshold-dominated left end of the curve has dropped by over a factor of three. For a 1 meter position and 3 degree orientation threshold the packet rate is only 1 3 packets per second. This is comparable to the packet rates generated by ground vehicles using the first order position, zero order orientation dead reckoning algorithm of Figure 2.

A more comprehensive comparison of simplest and most complex dead reckoning algorithms studied is presented in Figure 5 For each value of the thresholds, Figure 5 shows the ratio of the PDU transmission frequencies using the simple first order position, zero order

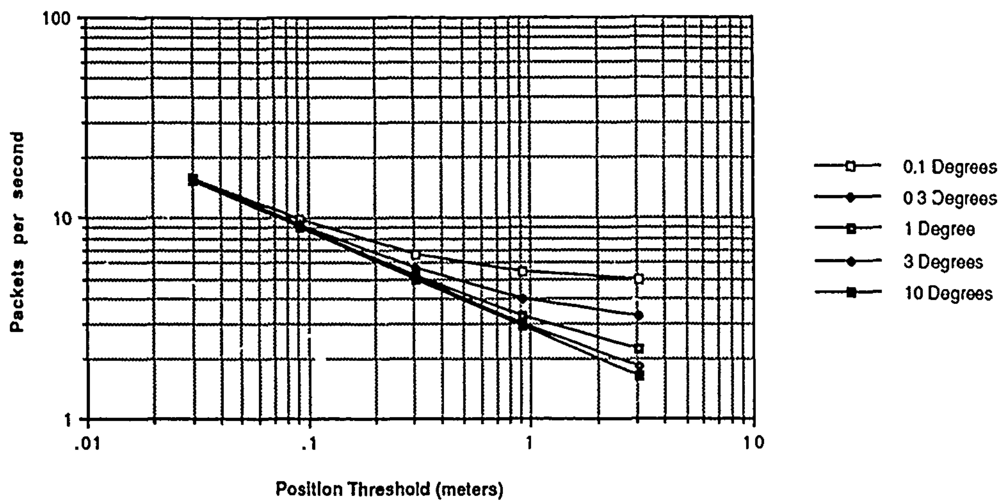


Figure 3: Algorithm 2: First Order Position, First Order Orientation

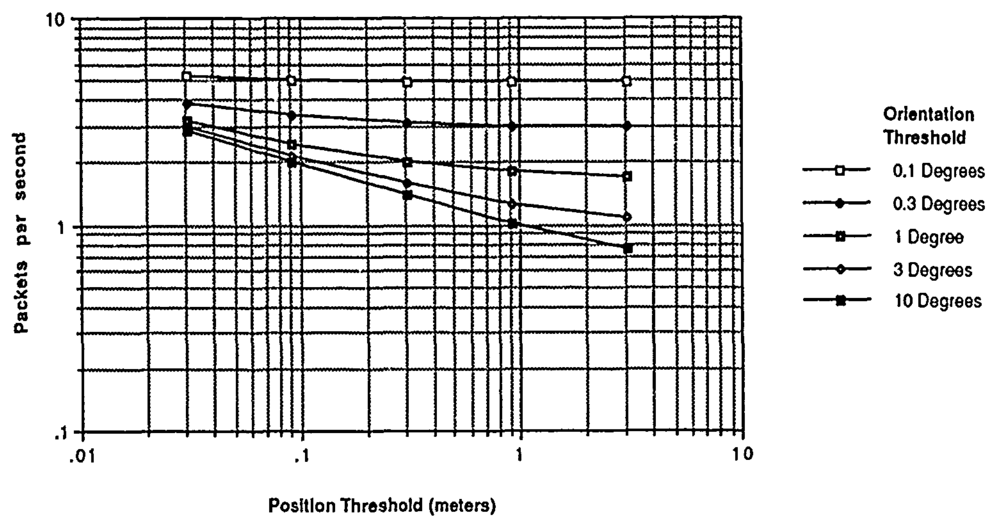


Figure 4: Algorithm 3: Second Order Position, First Order Orientation

orientation dead reckoning algorithm of Figure 2 to that obtained with the more complex second order position, first order orientation dead reckoning algorithm of Figure 4 The improvements in PDU transmission frequencies range from 300% to 700%. Note the pleasing result that the greatest improvements are achieved among the tightest thresholds.

Experiment Conclusions

Even the simplest possible dead reckoning algorithm can dramatically reduce the network bandwidth required to support a distributed simulation Additionally, even high performance aircraft simulators can achieve high position and orientation accuracies using only the slightly more complex first order position, first order orientation dead reckoning algorithm.

The use of a second order dead reckoning algorithm considerably further decreases the required network bandwidth, particularly for tight position thresholds. By choosing the dead reckoning algorithm to suit available computational and network bandwidth resources, high performance aircraft simulators with high fidelity position and orientation requirements may be incorporated in distributed simulations.

SUMMARY

Entity State PDU transmission frequencies encountered in this experiment are consistent with the results of a similar high performance aircraft simulation experiment using first and second order dead reckoning algorithms conducted by Kenneth D. Morris and Suresh Goel of Northrop Corporation. The results of the Northrop

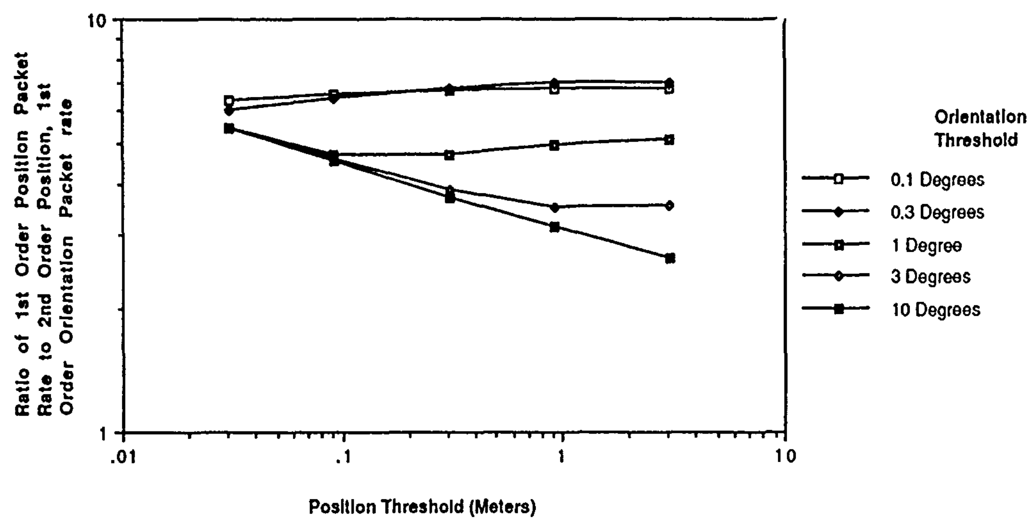


Figure 5: Comparison of Algorithms 1 and 3

PDU transmission frequency experiment and the BBN PDU transmission frequency experiment are consistent. DIS is fully capable of supporting high performance, high fidelity aircraft simulation. Although DIS may have been perceived by some in the simulation industry to lack the capability to provide sufficient fidelity for high performance aircraft simulation, the results of the two independent PDU transmission frequency experiments conducted to date show this is not the case.

The DIS networking standard is a robust, relatively simple, solution for networking hundreds to thousands of simulated entities in a common tactical environment. DIS provides a high validity tactical war fighting training medium which can cost effectively exercise all levels of command and control for U.S. naval, ground and air forces. Conduct of very large scale, joint operations simulation exercises will finally become possible in the 1990's as a result of the implementation of the DIS simulator networking standard.

REFERENCES

1. Schaffer, R., and R. Waters. "Dead Reckoning Algorithms and the Simulation of High Performance Aircraft", BBN White Paper ASD 91-015, UCF IST 4th Workshop on Standards for the Interoperability of Defense Simulators; March 1991

2. Miller, D., A. Pope and R. Waters. "Long-Haul Networking of Simulators," Proceedings: Tenth Interservice/Industry Training Systems Conference, Orlando, Florida; December 1989

3. Pope, A. and R. Schaffer, The SIMNET Network and Protocols, BBN Report 7627, Bolt Beranek and Newman Inc.; Cambridge, Mass.; June 1991

4. Military Standard (Draft), Entity Information and Entity Interaction in a Distributed Interactive Simulation, IST-PD-90-2 (Revised), University of Central Florida, Institute for Simulation and Training, January 25, 1991

5. Morris, K. and S. Goel. "Interim Report for Long Haul Data Transmission System"; Northrop Report NOR 91-28, Northrop Corporation, February 1991

ACKNOWLEDGEMENTS

The authors would like to thank General Dynamics, Flight Simulation Laboratory, Ft. Worth, TX for providing the F-16 Paris Airshow flight simulator data. We would especially like to thank Steve Norman, Doug Gould, Mike Lientz, Matt Landry, David Tedeschi, Ajay Patel, Becky Waren and Dwaine Robinson. The PDU transmission frequency experiment described in this paper would not have been possible without their help.

The initial studies reported in this paper were funded by the Air Force Human Resources Laboratory, Williams AFB. We would like to thank Major Michael Kamrowski, Research Engineer for the Aircrew Combat Mission Enhancement (ACME) - Multiship Network Development for his guidance and support.

The authors would like to thank Dr. Jerry Burchfiel and Rolland Waters of BBN for their helpful advice and assistance in conducting the studies reported in this paper.

ABOUT THE AUTHORS

Richard L. Schaffer is a staff scientist with BBN Systems and Technologies' Advanced Simulation Business Unit He holds a B.S. in Physics from Massachusetts Institute of Technology. For the past five years, he has worked on the SIMNET Large Scale Simulation project. Mr. Schaffer has also been actively involved in the Distributed Interactive Simulation standards process since its inception He is a member of two DIS working groups, and chairs a sub-group on articulated parts.

Edward P Harvey is a military aviation consultant with BMH Associates, Inc He holds a Chemical Engineering degree from Georgia Institute of Technology. Mr Harvey is a former F-14A Tomcat Radar Intercept Officer (RIO) and air combat maneuvering instructor with

1,400 hours of tactical fighter experience. He has been involved in a variety of flight simulation projects including the U.S. Navy F-14D Aircrew Training System, the DARPA/U.S. Army SIMNET program, and the DARPA/U.S. Navy WISSARD program. He is a member of the Distributed Interactive Simulation Time Critical working group.

APPLICATION OF THE SIMNET UNIT PERFORMANCE ASSESSMENT SYSTEM TO AFTER ACTION REVIEWS

Larry L. Meliza
U.S. Army Research Institute PM TRADE Field Unit
and
Seng Chong Tan
University of Central Florida Institute for Simulation and Training

Simulation Networking (SIMNET) provides a means to supplement collective field training, but research is needed to develop a SIMNET training strategy. The U.S. Army Research Institute (ARI) and Perceptronics developed a prototype PC-based Unit Performance Assessment System (UPAS) to collect time-tagged data on firing events and vehicle status from the SIMNET network and display data summaries. The UPAS is intended to assist in preparing unit performance summaries necessary to provide units with feedback during After Action Reviews (AARs) and conduct training research. This paper describes a project by ARI and the Institute for Simulation and Training (IST) involving the design and software implementation of procedures for combining network data with non-network data within the UPAS to support the preparation of improved AAR aids. These aids may be applied to future generations of networked simulators such as the Close Combat Tactical Trainer (CCTT).

INTRODUCTION

The networking of combat vehicle simulators makes it possible for crews to interact with one another on a common terrain database. Information produced by each simulator, such as its location on the terrain database and the target location of each firing engagement, is broadcast over the network and picked up by other simulators.

The initial simulation networking device (SIMNET) developed by the Defense Advanced Research Projects Agency included simulators for armor and mechanized infantry vehicles.[12] SIMNET, like all applications of Distributed Interactive Simulation (DIS), is intended to train crews to work together as part of a unit and train units to work together as part of a larger organization, but it is not intended to support individual skills training per se. SIMNET, for example, lacks the fidelity required to use it to train individual gunnery skills[4], but it can be used to train a unit how to use a volume of fire to cover the movement of another friendly unit.[2]

In 1988, Thorpe reported evidence that SIMNET training transfers to field training exercises.[12] In the interim, two efforts employing Armor Officer Basic students as subjects have provided additional evidence of training transfer[1,10], and three analytical efforts have identified collective tasks that might be trained in SIMNET.[2,4,6] Much of this training transfer work was accomplished in order to decide whether the relatively low level of fidelity of certain aspects of SIMNET prevent transfer to field training.

Three recent reports stressed the importance of assessing the effects of SIMNET feedback and practice variables on transfer of training.[1,6,10] Two of these reports provided evidence that the transfer of SIMNET training increases as trainers gain experience in the conduct of SIMNET training.[1,10] These two reports also suggest that the improved performance as trainers gain experience with SIMNET is a function of the quality of feedback given to exercise participants during After Action Reviews (AARs).

The AAR is not a critique of unit performance with respect to a predetermined list of standards. Instead, it focuses on events that contributed significantly to mission outcome and the causes of these events.[8,9] The AAR is intended to be an interactive learning process in which participants discuss what they did, why they did it, and possible alternative courses of action. The job of guiding discussions of events is made easier to the extent that information about these events is available in the form of graphs, tables, and figures.

In general there is a dearth of information regarding the AAR and practice variables that influence SIMNET training. Data on these variables are needed to develop efficient strategies for integrating SIMNET training into a total collective training strategy that includes field training exercises at home station and training at the Army's National Training Center (NTC).

The lack of information on SIMNET training is understandable when one considers the complexity of collective training combined with the lack of SIMNET data analysis tools. A collective exercise is generally composed of multiple interdependent collective tasks. How a unit performs one of these tasks influences; the conditions under which subsequent tasks are performed, whether or not certain tasks or subtasks must be performed, and how subtasks should be performed. This variability in unit performance requirements makes it necessary for trainers and researchers to perform substantial analyses to provide effective feedback and document the training that is conducted. It should not be too surprising that the various training transfer efforts have documented the training conducted only at a very broad level.

SIMNET includes powerful tools for observing unit performance during and after an exercise. These tools include a "Stealth Vehicle" that allows a trainer or researcher to obtain an "out the window view" of the action from any point on the battlefield and a Plan View Display that allows the action be observed from a bird's-eye view.[12] However, translating this wealth of available information to a format that supports documentation of training (practice and feedback) and measurement of unit performance is expected to be a substantial task.

THE PROTOTYPE UNIT PERFORMANCE ASSESSMENT SYSTEM (UPAS)

ARI and Perceptronics developed a low cost, personal computer-based (PC-based) Unit Performance Assessment System (UPAS) to assist in collecting and analyzing data from SIMNET exercises. UPAS collects virtually all of the data broadcast over the network for subsequent analysis on a stand alone basis. The prototype UPAS contains two types of tools to support training feedback and research.[14] First, UPAS loads data from the network into a relational database, and provides a menu-based system of editors for creating graphic and tabular summaries of unit performance from these data. The design of the database is based on the NTC database to support research on collective training strategies. Second, the prototype contains a Plan View Display (PVD) that can be used to replay the mission or critical segments of the mission from a bird's-eye view. In addition to showing vehicle location and weapon system orientation over a grid map, the PVD indicates when each vehicle fires or becomes a casualty. This prototype UPAS includes the capability to magnify the battlefield to the point where the entire display covers an area that is only one kilometer square. Figure 1 illustrates a PVD screen, and Figure 2 shows a graph developed with the UPAS.

In the next phase of development, ARI and the University of Central Florida Institute for Simulation and Training (IST) expanded and modified the UPAS to support training feedback and research more effectively, addressing two major concerns. First, it was necessary to begin integrating the data collected from the network by UPAS with data from non-network sources (e.g., unit plans for conducting the mission) to provide a more complete description of unit performance[5,6]. Second, it was necessary to provide data summaries that could be used to assess unit performance quickly after an exercise. Unlike a field training exercise, there are few post-mission tasks after a SIMNET exercise to keep a unit occupied while a trainer analyzes unit performance in preparation for providing feedback. It is critical that UPAS support the preparation of timely AARs. Figures were considered to be a good vehicle for integrating network data with non-network data and providing quick summaries of unit performance.

APPROACH

ARI developed concepts for Plan View Display modifications and new types of AAR aids to integrate network and non-network data while providing easily interpretable descriptions of unit performance. These concepts were further specified by examining a sample of unit performance measures that might be addressed using these aids to identify the information each type of aid would need to contain. The sample measures were Armor Platoon Mission Training Plan standards[3] classified as being supported by SIMNET[2] that might employ the network data collected by UPAS.

The second step was to develop the software necessary to implement the concepts of new or improved AAR aids. One challenge that faced IST was to implement these concepts in a manner that supports rapid display of each aid after an exercise. A second challenge was to implement these aids in a manner that allows them to be used or modified in a flexible manner.

The third step was to categorize representative measures of unit performance according to similarities in network and non-network data requirements necessary to support their application. The fourth step was to assign categories of performance measures to AAR aid formats based upon each format's ability to accommodate data types. This step was necessary to facilitate systematic application of AAR aids to performance measurement.

The results are described below in terms of; the AAR aids implemented, the software techniques employed, and the categories of performance measures appropriate to each aid.

NEW UPAS AAR AIDS

The PVD was modified to support training feedback and research more effectively, and three new types of AAR aids were implemented. Two of the new aids, the Battle Flow Chart and Battle Snapshot, provide a bird's-eye view of the battlefield conveying information that is not easily addressed by the PVD format. The third aid is in the form of a timeline.

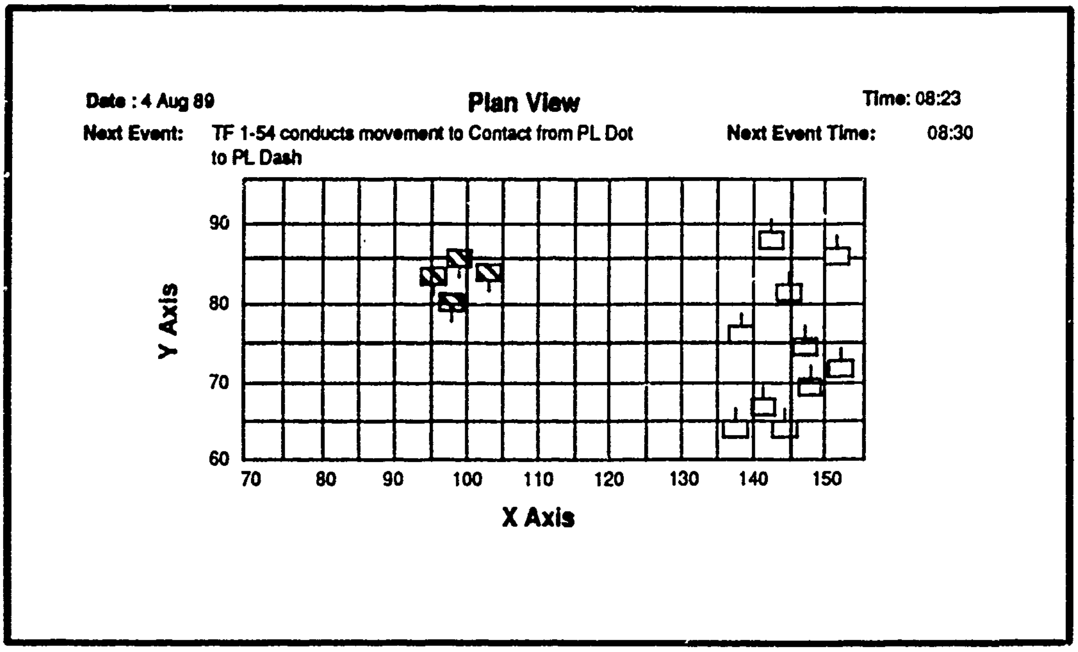


Figure 1. Example of a UPAS Plan View Display (PVD) Screen

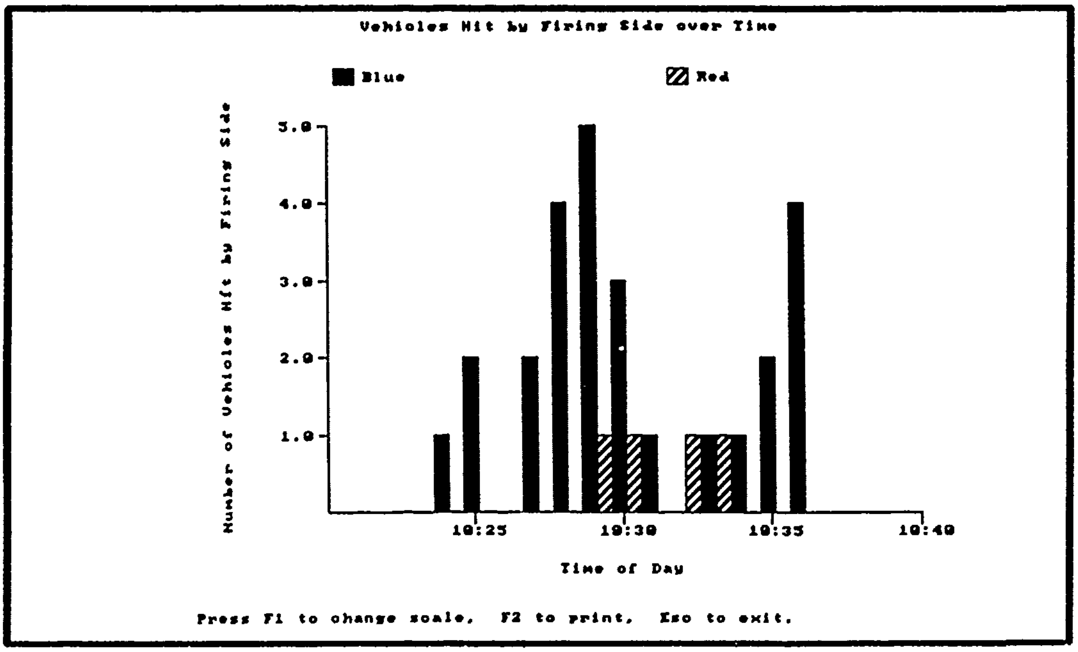


Figure 2. Example of a graph prepared by the UPAS showing volume of fire as a function of time and unit side.

The capability to display major terrain features (i.e., highways, unimproved roads, tree lines, tree canopies, buildings, and bodies of water) was added to the Plan View Display as shown in Figure 3. These features are all color coded in the display. In addition, a quick search capability was implemented to allow the user to move quickly forward and backward to particular points in an exercise, and the capability to magnify the battlefield was enhanced to allow sections as small as 200 meters squared to be displayed.

The Battle Snapshot shows the position and orientation of vehicles and weapon systems from a bird'seye view (See Figure 5). A Snapshot can be taken of any point in the exercise designated by the user. Like the PVD and Battle Flow, the Snapshot employs a grid map that includes terrain features and control measures.

....An Exercise Timeline is a tool for looking at temporal coordination of movement, firing events, control measures, and communication. An example of a timeline is provided in Figure 6. The top and bottom lines cover the time during the

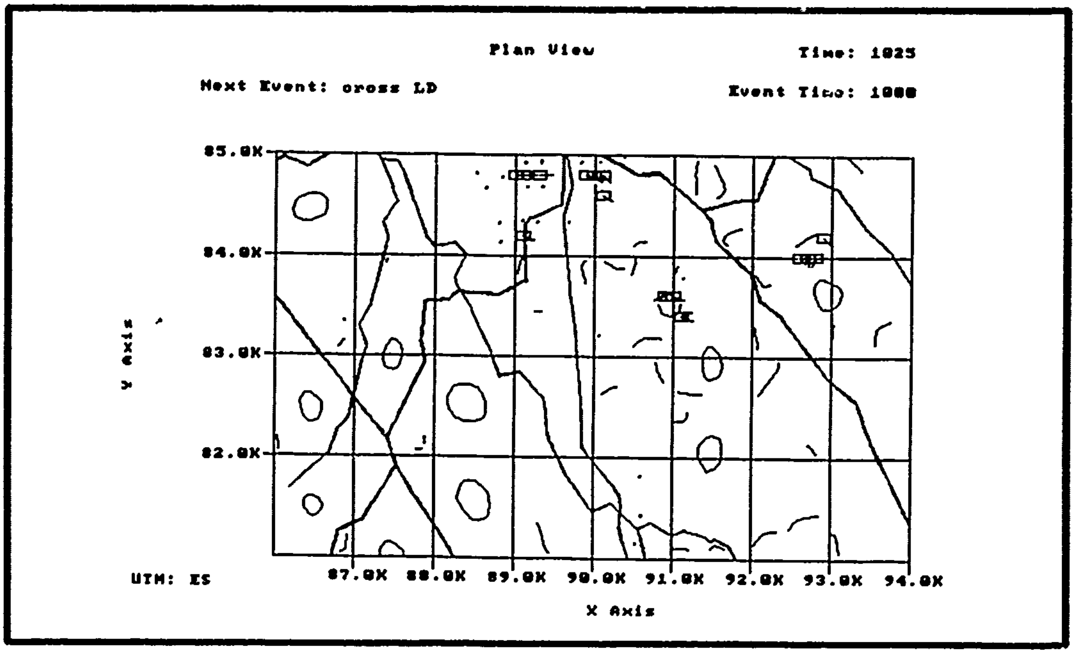


Figure 3. Improved UPAS Plan View Display

A Battle Flow Chart was implemented to trace the movement of vehicles from a bird's-eye view at specified intervals throughout the course of a mission (see Figure 4). The Battle Flow traces movement over a grid map displaying terrain features and unit control measures. The user can start movement at any point in the exercise, and thus a hard copy of the trace can be made for selected portions of the exercise as well as for the entire exercise. The Battle Flow indicates vehicle location but does not indicate vehicular and weapon system orientation. The Battle Flow, like the Plan View, allows the user to magnify the battlefield.

exercise. The second line describes movement of the platoon as a function of time and unit control measures by using bars to indicate the time when the first and last vehicle of a unit crossed a control measure. The Timeline also indicates the beginning and ending of periods of time when the entire platoon was halted. The third line provides information about the time of direct and indirect firing events. A small square is used to indicate a point in time when the unit receives artillery fire, and a down arrow indicates when the first enemy direct fire was received by the unit. A circle indicates when a friendly vehicle is destroyed. An up arrow indicates when the unit first delivers fire on the enemy, and "x" indicates when an enemy vehicle is destroyed.

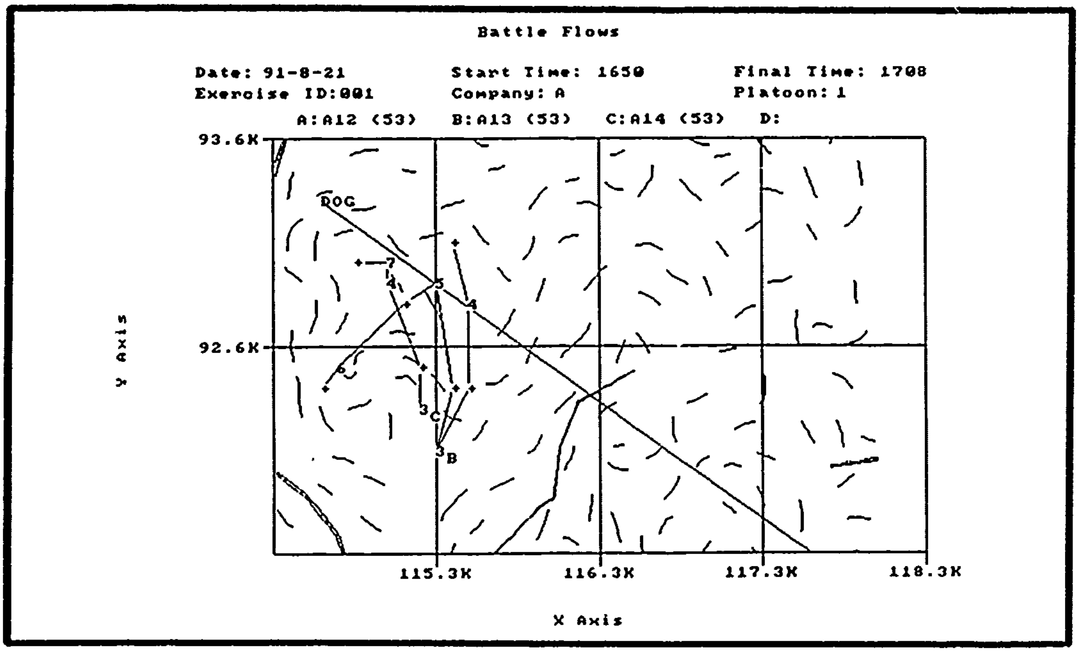


Figure 4. Sample Battle Flow Chart.

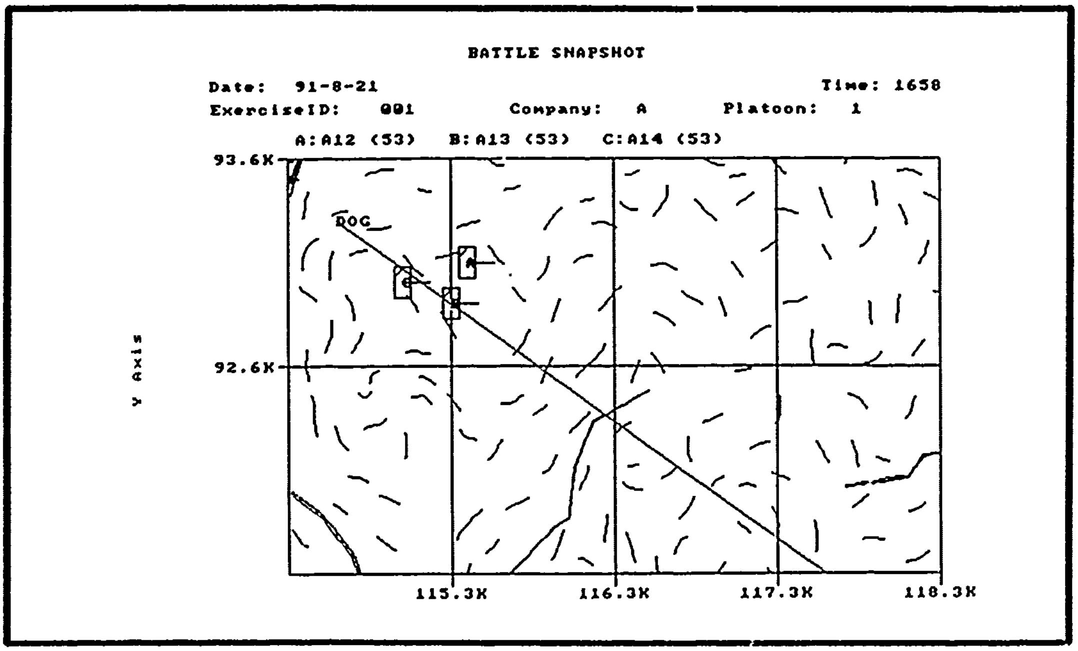


Figure 5. Sample Battle Snapshot.

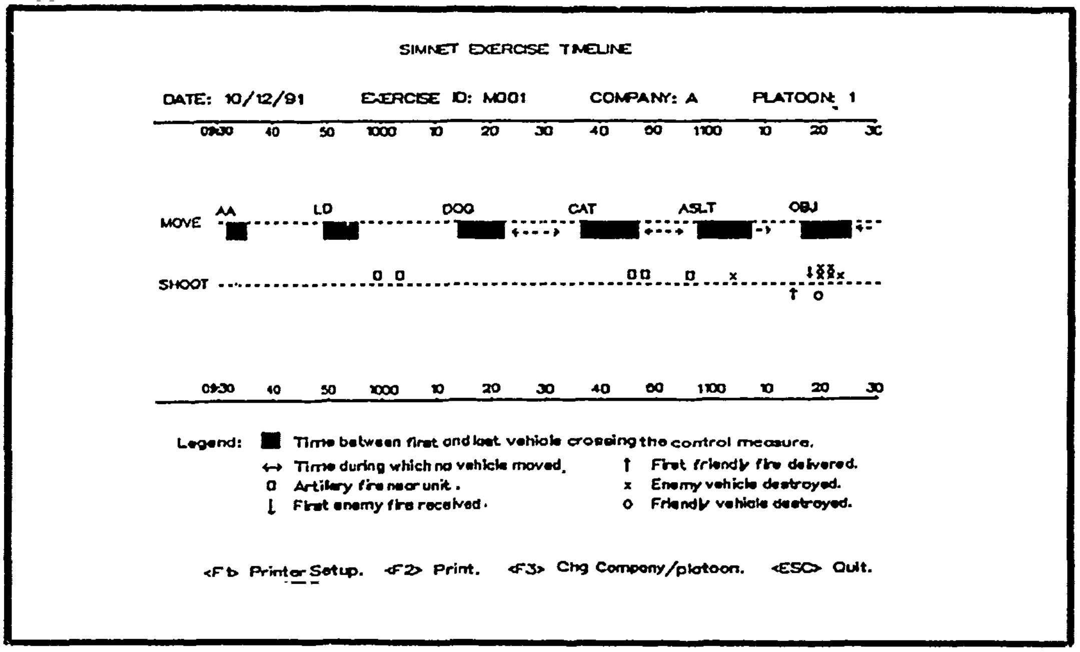


Figure 6. Sample SIMNET Exercise Timeline.

PROGRAMMING METHODS USED TO IMPLEMENT AAR AIDS

Integration of Non-Network Data Sources

Unit control measures, terrain data, and tactical communications are three of the major sources of non-network data to be integrated with network data to support performance assessment. How well movement, firing events, and communications comply with control measures is an important measure of unit proficiency. The manner in which a unit adjusts its actions to reflect the terrain situation is also an important measure of unit proficiency. Further, terrain is important because the unit's vision, movement, and firing activities are, to a large extent, constrained by the surrounding terrain features. Finally, tactical communication data are critical because SIMNET's greatest strength is estimated to reside in the ability to support the training of command, control and communication activities within units.

A utility was developed to allow the user to input information about the name, type, and location of control measures into the relational database. UPAS currently supports any two dimensional control measure that can be represented by a point, line, or circle. These control measures are displayed in the PVD, Battle Snapshot, and Battle Flow chart, and they are used by the Exercise Timeline to compute when a platoon crosses each control measure. Future work in this area involves integrating three dimensional control measures, such as Airspace Coordination Areas.

UPAS has successfully incorporated terrain features into the PVD that include: tree lines, tree canopies, dirt roads, paved roads, rivers, and buildings. All information is obtained from the SIMNET terrain database that includes a database header and a list of terrain patches. A copy of the terrain database is loaded into the PC, and in the case of the SIMNET database for Fort Knox, it requires approximately 32 mega-bytes of memory. The Ft. Knox terrain database is composed of 15,000 terrain patches, with each patch representing a 500m x 500m square of land. Associated with each terrain patch is information about vertices, edges, terrain polygons, trees, tree lines, objects, canopies, and its coordinates. UPAS uses these coordinates to determine which patches to retrieve from the database for display on the PVD screen. Future work in this area involves adding contour lines to the PVD.

Tactical communications data still need to be integrated with the other data sources. Software will be developed for for integrating communications data with movement and firing event data in the exercise timeline. Future networked simulators will employ methods for digital transmission of radio messages over the simulation network , allowing tactical communications to be picked off the network and loaded into the relational database in the same manner as other network data packets. For SIMNET, software will need to be developed to allow observers to input data on monitored tactical communications into the relational database.

Speed

One of the most critical problems encountered was the large amount of time required to apply the PVD and Battle Snapshot to unit performance assessment. Effective use of these aids requires moving forward or backward quickly from one point in the battle to another. The prototype aids moved only forward in time, and many minutes were expended in moving from one point in the battle to another. The slow movement was due to the fact that these aids work by reading sequentially the series of numbered data packets to locate the time of interest to the user. For example, in moving from 1000 hours to 1015 hours, the program looks at the time stamp for each intervening data packet until it finds a packet displaying 1015. This problem was addressed by implementing a utility that creates an index file containing packet addresses whose time stamps are one minute apart from each other. When the user selects a new time to move to, the program uses the time to retrieve the appropriate packet address from this index file, and then it uses the packet address to retrieve the appropriate packet from the raw data file. In this way, only two disk accesses are used to find the desired point in the battle.

After integrating the terrain database with the PVD, a new problem was encountered regarding the time required to access a display. A substantial amount of time was required to generate the terrain display for the initial Plan View screen. This screen covered a 16 by 8 kilometer segment of the battlefield containing information from 512 terrain database patches. This problem was addressed by reducing the display to an 8 by 4 kilometer area covering only 128 patches.

Flexibility

Another important problem addressed was that of creating a system that could be adapted by the user. This flexibility was achieved, in part, by incorporating display options. For example, the UPAS user has the option of selecting the frequency with which the locations of vehicles are displayed in the Battle Flow Chart. The time dimension has been annotated on the flow path of each vehicle by placing position update markers on the path which are spaced at an interval selected by the user. This feature allows the user to choose a larger position update interval for a larger (or longer) exercise to avoid over-cluttering the path with too many markers.

Future work in this area is directed towards two subgoals. The first subgoal is to define the requirements for modifying the AAR aids to support performance assessment above platoon level. The second subgoal is to decide if there is a need to make the UPAS graph and table editors more flexible. The present graph editor is limited to producing graphs that display information as a function of time and counting variables only, and the present table editor will not support the preparation of tables that involve more than one Structured Query Language (SQL) query.

CATEGORIES OF PERFORMANCE MEASURES APPROPRIATE TO EACH TYPE OF AAR AID

Table 1 indicates the categories of measures assigned to each type of AAR aid. Categories involving communication require utilities for displaying tactical communications in the Exercise Timeline before they can be fully supported.

TABLE 1. CATEGORIES OF STANDARDS APPROPRIATE TO EACH TYPE OF AAR AID

CATEGORY OF STANDARD	AAR AID FORMAT				
	SCORECARD	FLOW CHART	SNAPSHOT	PVD	TIME-LINE
MOVEMENT AND FIRING				x	x
FRIENDLY AND ENEMY FIRES	x			x	
MOVEMENT AND CONTROL MEASURES		x	x	x	x
MOVEMENT TECHNIQUE AND METT-T		x	x	x	
MOVEMENT AND COVER/CONCEALMENT		x	x	x	
WEAPON ORIENTATION			x	x	
HALTS AND COVER/ CONCEALMENT		x	x	x	x
LOCATIONS OF FRIENDLY IDF MISSIONS AND ENEMY POSITIONS	x				
SPATIAL RELATIONSHIPS AMONG MOVING VEHICLES		x	x	x	
RATE OF MOVEMENT				x	x
LOCATION, CONTROL MEASURES AND COMMUNICATIONS					x
FIRING EVENTS AND COMMUNICATIONS					x

RELEVANCE OF UPAS TO OTHER APPLICATIONS OF DISTRIBUTED INTERACTIVE SIMULATION

UPAS has the potential to serve as a research tool in developing feedback systems for a variety of DIS applications. The Army's future fielding of networked simulators will comply with standards for interoperability of defense simulations, beginning with the Close Combat Tactical Trainer (CCTT). These standards include specification of the content and format of data broadcast over the network. The UPAS is designed to support current SIMNET protocols that differ from those in the interoperability standards. However, the functional requirements for the CCTT require interoperability with SIMNET, necessitating the development of a translator between the standard protocol and the SIMNET protocol. This translator will make it possible for the UPAS to

address data packets from future DIS applications as well as supporting the current SIMNET.

The training research conducted on SIMNET using UPAS should also provide useful information for the development of other DIS applications. This information may include techniques for organizing, analyzing, and summarizing performance data, and it may include information relevant to the development of DIS training strategies.

SUMMARY AND DISCUSSION

UPAS is designed to support the interrelated goals of providing feedback to units and testing the effectiveness of SIMNET under various training strategy options. We have placed special emphasis on AAR aids that can be used to provide units with feedback promptly after an exercise. Future efforts will be concerned more with implementing performance measures that may require too much time to apply to fit into the short suspense framework of the AAR.

The UPAS is designed to be flexible enough to allow trainers and other system users to modify feedback displays to accommodate new information needs as they are discovered. This flexibility makes the UPAS an efficient tool for research on how to provide feedback.

The next step in UPAS development is testing the effectiveness of UPAS AAR aids as training feedback and research tools. The majority of this work will be completed at the Fort Knox, Kentucky SIMNET-Training Facility.

REFERENCES

[1] Bessemer, D. (1990). Transfer of SIMNET Training in the Armor Officer Basic Course (ARI Technical Report 920). Alexandria, VA: U.S. Army Research Institute for the Behavioral and Social Sciences.

[2] Burnside, B. (1990). Assessing the Capabilities of Training Simulations: A Method and Simulation Networking (SIMNET) Application (ARI Research Report 1565). Alexandria, VA: U.S. Army Research Institute for the Behavioral and Social Sciences.

[3] Department of the Army (1988). Mission Training Plan for the Tank Platoon (ARTEP 17-237-10-MTP). Washington, DC: Author.

[4] Drucker, E. H. and Campshure D. A. (1990). An Analysis of Tank Platoon Operations and Their Simulation on SIMNET (ARI Research Product 90-22). Alexandria, VA: U.S. Army Research Institute for the Behavioral and Social Sciences.

[5] Hiller, J. H. (1987) Deriving Useful Lessons from Combat Simulation. Defense Management Journal. Second and Third Quarter, 28-33.

[6] Kerins, J. W., Atwood, N. K., & Root, J. T. (1990). Concept for a Common Performance Measurement System for Unit Training at the National Training Center (NTC) and with Simulation Networking (ARI Research Report 1574). Alexandria, VA: U.S. Army Research Institute for the Behavioral and Social Sciences.

[7] McDonald, L. B., Pinon, C, Glasgow, R., and Danisas, K. (1990). Military Standard (Draft): Entity Formation and Entity Interaction in a Distributed Interactive Simulation. Orlando, FL: Author.

[8] Meliza, L. L., Sulzen, R. H., Atwood, N. K., & Zimmerman, R. (1987). National Training Center Guidebook for Force-on-Force Company/Platoon/Observer/ Controllers. Fort Irwin, CA: National Training Center.

[9] Scott, T. D. & Fobes, J. L. (1982). After Action Review Guidebook I: National Training Center (ARI Research Product 83-11). Alexandria, VA: U.S. Army Research Institute for Behavioral and Social Sciences.

[10] Shlechter, T. Bessemer, D., & Kolosh, K. (1991). The Effects of SIMNET Role Playing on the Training of Prospective Platoon Leaders. ARI Technical Report (in preparation). Alexandria, VA: U. S. Army Research Institute for the Behavioral and Social Sciences.

[11] TEXCOM Combined Arms Test Center (1990). Close Combat Tactical Trainer (CCTT) Force Development Testing and Experimentation (FDTE) (TCATC Test Report FD 0200). Fort Hood, TX: Author.

[12] Thorpe, J. A. (1988) Warfighting with SIMNET-A Report from the Front. pages 263-273. Proceedings of the 10th Interservice and Industry Training Systems Conference.

[13] U.S. Army Armor School (1990). Training Device Requirement (TDR) for the Close Combat Tactical Trainer (CCTT). Fort Knox, KY: Author.

[14] White, S., McMeel, K., & Gross, W. (1990). Unit Performance Measurement and Training Program Design for Networked Simulators (PTR-4013-1-90). Woodland Hills, CA: Perceptronics.

ACKNOWLEDGMENT

The authors wish to thank LTC Thomas Mastaglio, Headquarters, U.S. Army Training and Doctrine Command, for his review of the draft of this paper.

ABOUT THE AUTHORS

Larry Meliza is a Research Psychologist with the U.S. Army Research Institute, PM TRADE Field Unit. He holds a Ph.D. in Psychology from the University of Arizona. His experience in collective performance

measurement and feedback includes field testing applications of engagement simulation to collective training, developing prototype Army Training and Evaluation Program (ARTEP) Mission Training Plan documents as part of the ARTEP improvement effort, and developing guidebooks for Observer/Controllers at the U.S. Army's National Training Center (NTC). He is currently involved in the design of a SIMNET Unit Performance Assessment System. In addition, he provides technical advisory service to PM TRADE regarding future generations of SIMNET.

Seng Chong Tan is a Software Engineer with the Institute for Simulation and Training. He holds a Master of Science degree in Computer Science from the University of Arizona. His experience includes the development of software for filtering, storing, managing, and retrieving network data across a variety of applications. At present he is principal investigator for an effort to filter SIMNET data and develop After Action Review Aids for the SIMNET Unit Performance Assessment System.

OBJECT-ORIENTED ANALYSIS: THE TRANSITION FROM REQUIREMENTS ANALYSIS TO DESIGN

Jerry H. Hendrix
Boeing Defense and Space Group
Simulation and Training Systems
Huntsville, AL

ABSTRACT

Software development in today's complex technological society is becoming the "tent-pole" for trainer program slides and cost overruns. The Department of Defense has recognized this crisis and built stringent software standards such as DoD-STD-2167 and DoD-STD-2167A and required a common computer software language. Contractor approaches are needed to minimize software development complexity and provide the training customer with a concise software development methodology to produce concurrent training equipment. The next generation of training equipment, built concurrently with the aircraft, such as ATF and LH trainer programs, must provide a thorough software development approach to provide maximum reusability of software from the air vehicle to the training equipment.

This paper discusses an approach that allows simplicity when building software in a complex environment. It presents a way of allowing software contractors to develop simpler, more precise software specifications that map directly to preliminary software designs. For many years, software developments have focused on the transition between design and implementation due to inadequate software development methodologies and software languages that don't portray design understanding. Specifications have been written too early in development, resulting in the loss of all mapping of requirements to design. During implementation phases, developers lose sight of program and software requirements. The use of object-oriented principles, automated tools, Ada, and flexible company standards can result in flexible, concurrent software projects. Specifically, this paper will give a candidate approach to transition from object-oriented requirements analysis to object-abstracted design within the guidelines of DoD-STD-2167A principles.

INTRODUCTION

Present day training contracts are contained within the prime hardware procurement for that tactical weapon system. Several modern programs, such as the Light Helicopter (LH), Advanced Tactical Fighter (ATF), Non Line of Sight (NLOS) missile system have required the procurement and development of the training systems as part of the prime contract. Several problems arise due to this new acquisition strategy. These problems include common hardware and software development and the usability of software in different applications (aircraft, trainer, laboratories), and across different companies within the prime contract, development of embedded training equipment (who produces it? The tactical hardware vendor or a subcontracted training house?), internal prime contractor management approaches, and prime contract teammates hardware and software development approaches. For the innovative prime contractor, these problems become opportunities which lead to cost savings, performance increases, and quality products. As part of specifying the training equipment procurement within the prime contract, the government has required the use of the Ada computer language and DoD-STD-2167A. This paper addresses several problems dealing with software development in the training equipment buy-through-prime environment. This paper will also present an object-oriented approach to software development that will facilitate common software across the entire weapon system. Those who are a part of a prime contractor team building training equipment will benefit from this approach.

ADA LANGUAGE USAGE

The use of the Ada language offers an opportunity to develop software that is portable and usable across several sets of processing equipment. The Ada technology offers the capability to enforce the developmental process by providing mechanisms to ensure a consistent, localized, modifiable, and maintainable product. The Ada technology also provides mechanisms for handling data voids without impacting the developmental process. The classical waterfall process of performing concept definition, requirements analysis, design, implementation, and test can be enhanced through the usage of the Ada technology combined with object-oriented analyses and prototyping. The enhancement is a "blurring" between phases in which successor phase issues are addressed in predecessor phases, thus providing mechanisms to solve classical design and integration problems early in the development process.

Many developmental approaches (Object-Oriented Design, etc.), and variations on those approaches, have been used for Ada software development. Some approaches are applicable for real-time software systems and some approaches are better suited for those non-time critical functions. For weapon system development, a unified approach for the weapon system software development is by far the most important criteria for development of objects.

OBJECT-ORIENTED ANALYSIS

Object-oriented analysis is not a magical, mystical art; it is an analysis that everyone performs everyday. For example, we answer The_Phone, we drive A_Car, we wear Clothes, and in the training industry we model The_Device_to_be_Simulated. In an aircraft trainer, for example, if one thinks of the objects, he thinks of Flaps, Ailerons, Engines, Tires, etc. Objects can be divided into three major categories: 1) Physical, Tangible Objects (Pumps, Flaps), 2) Nature or Laws of Nature (Atmosphere, Equations of Motion, Bernoulli or

Maxwell's equations), and 3) Group of operations that provide manipulation of objects (Math Utilities, Overloaded Operations). Objects can be determined via several means: Brainstorming sessions with systems engineers, software designers and hardware designers, analyses of hardware drawings, analyses of the environmental requirements (types of atmospheric effects, countermeasures, threats), and analyses of operational requirements (2 v 4 operations, procedures training) all yield object lists. One of the hardest parts of any object-oriented analysis is to understand at what level one stops identifying hierarchical relationships of objects. The electrical system within a device is an object, whereas generators are an object and individual parts of a generator down to minute circuitry are objects. At what level does one stop decomposing objects? The answer to this question is dependent upon the fidelity of the system requirements. On the shuttle trainer, requirements dictated that modeling needed to go down to the circuitry/wiring diagram level. That simulation lead to NASA ultimately saving the life of a shuttle crew. On a cockpit or iron bird type of procedures trainer, detailed modeling of a particular system is not required. The Instructional Systems Development (ISD) helps yield the type of task and trainers needed for a particular weapon system. One thing to point out about common object development is that parent composite objects can always be decomposed into children.

SELECTION OF COMMON OBJECTS

Common objects are software objects that can be ported and reused in other parts of a program or project. Common objects include common objects purchased, common objects developed within a software development team, and common objects shared across the project. Purchased common objects are usually objects which provide operations that manipulate other objects (math utilities) or are laws of nature. Common objects within a software development team are those objects unique to that development activity and are design based on project unique requirements. For example, a generic pump model can be developed to address malfunctions needed for training unique simulations. Those malfunctions may not be needed in a pump model in which high fidelity is not required. Considering the training equipment buy-through-the-prime strategy, several opportunities arise to provide common software across the project.

While common software objects used across projects provide the capability for lower software development cost and shorter development schedules, several problems arise when objects are shared across the project. Potential problem areas for common software development are:

•1) Determination of common objects that meet multiple requirements;

•2) Who develops the common object;

•3) Logistics of the development schedule to support all users; and

•4) Target platform(s) issues.

One key strategy of acquisition through the prime must be the sharing of common software components.

DETERMINATION OF COMMON OBJECTS

Common object definition requires a concerted, unified effort of a joint software engineering working group. This working group must be comprised of product groups that share common needs. These product groups may include avionics, laboratory integration, and training software development groups. Key issues to be resolved by the working groups include: Identification of software that is developed by particular product groups, which can be used completely

by other product groups. For example, Operational Flight Programs (OFPs) are developed and maintained by the avionics group and the training equipment developers are users (the training equipment developers stimulate the OFPs). Identification of software that is co-developed by groups with similar requirements are candidates for common software. For example, avionics integration requires the use of models that represent the environment surrounding OFPs. The training equipment could benefit from this type of software, as long as it is built to training unique requirements. The development of these common objects requires key developers from each product discipline to participate in the development.

A core group of developers may be utilized to develop common objects. Key members include systems engineers and software engineers from product areas that can benefit from common software. Each product area engineer must be familiar with the product and its requirements. Common software development and coding standards must be developed within the program to provide maximum utilization of common objects. Once common objects are identified, they must be prioritized and scheduled based on program product need. One product may need the common object well in advance of another product. It may be beneficial to the problem for the later product management to allocate resources to the common object development. This may pose unique manloading curves for product disciplines or across project teammates.

Additionally, the target platform for common objects must be considered when developing common objects. The portability guidelines published in many books assist in addressing this issue.

REQUIREMENTS ANALYSIS: OBJECT-ORIENTED AND 2167A

Trying to specify the behavior of a large weapon system is not a trivial task! As customer needs increase due to threat and threat theater changes, the technique of modeling behavior must provide a flexible, yet controlled, system representation. A top level behavior analysis will quickly yield objects throughout the system. From this analysis, the system must be broken into manageable segments, prime items, configuration items, DoD-STD-2167A requirement capabilities (functions), DoD-STD-2167A Computer Software Components (CSCs), and DoD-STD-2167A Computer Software Units (CSUs) (FIGURE 1). CSCs and CSUs are system design representations, whereas capabilities are system behavior representations. The two representations are not required to match, but the design must behave according to behavioral representations.

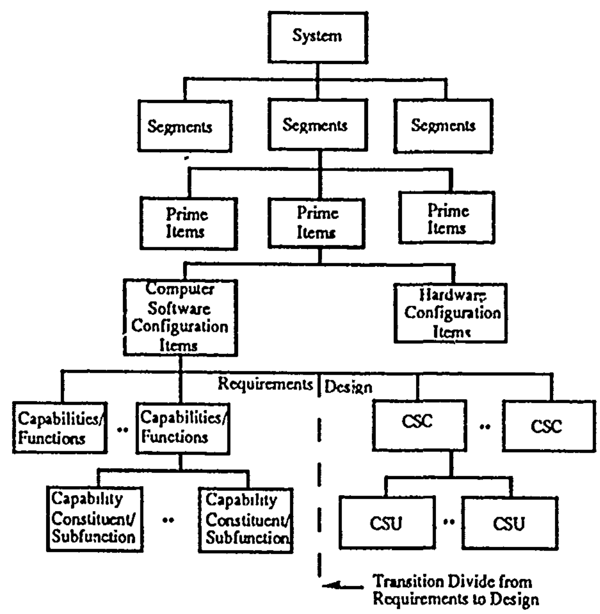


Figure 1
System Breakdown

Object-oriented analysis begins at the segment level. Several graphical tools are commercially available that can assist in analyzing system behavior and leading to the determination of system objects. These analyses lead to the segment characteristics. Characteristics are object specificity based on the segment. In the Space Station Freedom Program (SSFP), the Environmental Control and Life Support System (ECLSS) has multitudes of pumps to regulate air supply. The Ada language will be used to represent the definition of pumps in a compilable Ada format. Ada is used in several ways during requirements analysis:

- Universal operations specification
- Segment characteristics specification

Universal operations is a collection of universal constants and manipulation operations that are needed by this segment. Examples include PI = constant 3.1415 or whatever accuracy is needed; Two PI = constant PI * 2.0, etc. The universal constants and operations are defined in an Ada package by that name.

REQUIREMENTS DESIGN LANGUAGE (RDL)

Segment characteristics are the identification of the device to be modeled or built. These characteristics are specified in an Ada package. This Ada package provides, a physical representation of the segment from a software perspective. For example, if one were to need to model the propulsion effects on a Boeing 767, the characteristic representation would be like that shown in FIGURE 2 . This specificity requires true systems engineering and analysis work to be performed. Design data and schematics must be thoroughly analyzed to specify the segment in Ada RDL.

```
package Boeing_767_Aircraft_Characteristics is

  type Engines is (Left_Engine, Right_Engine);

  type Thrust is Real;

  type Propulsion_Effects is array (Engines) of Thrust;

end Boeing_767_Aircraft_Characteristics;
```

Figure 2
Segment Characteristics Example

Once the segment objects are identified, relational analysis begins in the context of CSCIs. Objects are grouped to form CSCIs. The pumps, fire detectors, and smoke detectors, etc., form the Fire Detection and Suppression CSCI of the SSFP ECLSS. Relational analysis follows requirements allocations to the segment and CSCI from the system requirements. These allocations set the context of the object determination and definition and the relational analysis. Understanding how objects relate leads to the specificity of interfaces. These interfaces between CSCIs and objects are "prototyped", using compilable Ada RDL. This technique is valuable in large programs because the interface specificity is consistent. Thus, consistent interfaces are used to manage/enforce software connectivity with subcontractors and within.

Relational analysis continues inside the context of the CSCI. Understanding relations between objects leads to the identification of internal CSCI interfaces. The requirements allocated to this CSCI are extended to specify processing requirements. During object-oriented requirements analysis, capabilities identified are system objects. These objects correlate directly to design CSCs. See Figure 3. Requirement capabilities correlate to CSCs and

requirement constituents correlate to CSCs or CSUs. Object capabilities and CSCs consistency provide maximal understanding by customers, thus leading to a program that eases review hassles.

During requirements analysis, the software structural model must be defined. The structural model provides the architecture format by which CSCs will be mapped to Ada constructs. This provides consistency in the Ada product since all Ada package formats would be the same for each object (CSC). This consistency enhances reusability, maintenance and modification tasks. The structural model is a necessary part of the software requirements task.

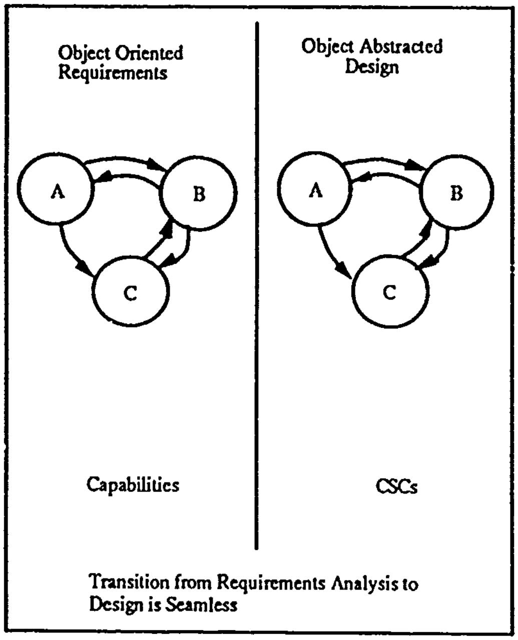


Figure 3
Capabilities = CSCs

DoD-STD-2167A provides enough flexibility for object specification. Object-oriented requirements analysis can map well to a DoD-STD-2167A Software Requirements Specification (SRS). The SRS data item description provides the mechanism to document object-oriented requirements analysis. Since in object-oriented requirements analysis the Ada language is used to represent some requirements, questions arise as to the type of requirements and detail content one would put in the SRS. Many object-oriented design advocates encourage requirements in the SRS that provide design implementation detail. The following describes SRS sections and the object-oriented requirements that can be found in SRS sections. Following this mapping will be warnings to observe when implementing object-oriented requirements analysis and documenting them in an SRS.

SECTION 3.1 CSCI EXTERNAL INTERFACE REQUIREMENTS

Section 3.1 of a DoD-STD-2167A Software Requirements Specification identifies the CSCI external interfaces. Each interface is to be identified by name and brief description. In this section, the first benefit from using the Ada language in requirements analysis is obtained. The interfaces can be represented by a set of compiled CSCI (Ada objects) specifications. Each compiled specification, combined with other compiled CSCI specifications, provides a consistent set of external interface requirements. The strategy employed is that CSCI to CSCI design interface requirements are fulfilled in these specifications. The set of Ada code specifications can be provided in section 6.? Notes of the SRS. Section 3.1 would ... in tabular form or pictorial form ... output of the interface analysis performed using Ada as a requirements definition language. The advantage of using Ada for interface specificity is the consistency and checking inherent when compiling Ada specifications, and a bounding of the design solution.

SECTION 3.2 CSCI CAPABILITY REQUIREMENTS

This section is required to identify all capability requirements that the CSCI must satisfy. In object-oriented requirements analysis, the capabilities should match the design CSCs or the design CSCs should be decompositions of the capabilities. If constituent capabilities are identified, they match to lower CSCs or CSUs. Section 3.2 also requires an identification of each system state and mode and the correlation of capabilities to those states and modes. This part of section 3.2 can be represented from part of the segment characteristics Ada package. The requirements represented by English text should define the actions, operations, and performance requirements of the capability. These requirements correlate directly to operations of the object and therefore relate to Ada subprograms with object packages in design. This relation provides a "seamless" transition from requirements analysis to design.

Figures 4 and 5 describe pictorially the process of object abstracted development in requirements analysis and design and the Ada language usage

The aforementioned discussions and figures represent an approach to facilitate "teamwork" between the software requirement specifiers and the software designers. The content identified for each section of the SRS provides the level of detail needed by customers to evaluate requirements and also bonds the design with requirements without specifying design. This strategy facilitates larger design participation during requirements analysis, but captures the content of object-oriented analysis in a manner by which the requirements analysis does not change scope. If this strategy is implemented, the requirement analysis phase expands over classical programs, whereas the design phase contracts. This approach also will not expand the test phase.

WARNINGS, OBSERVATIONS, AND CONCLUSIONS

Many organizations developing software in today's complex software world dare to believe in anything that contrasts to the "way we have always done software". Let us state that there are many viable ways to produce software; some focus on real time operation, some focus on the "ilities" of software (Maintainability, Modifiability), some focus on database development, and some are traditional company ways. Considering the experience gained in the Ada community today, many are saying that some sort of object-oriented approach is the methodology of choice. This paper has presented one viable, proven approach for the generation of an object-oriented Ada system. The following is a list of warnings to observe in this rapidly changing object-oriented world.

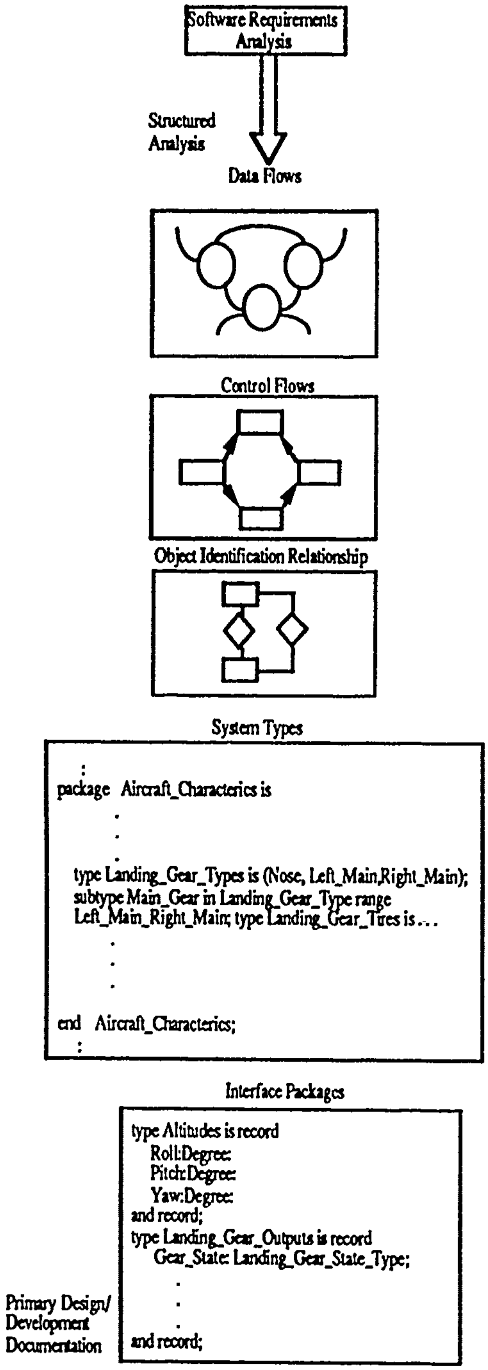
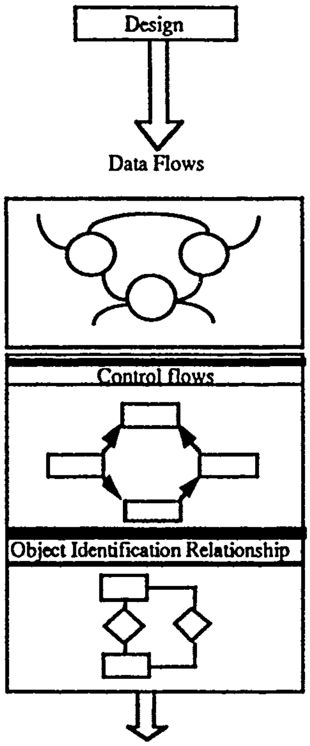


Figure 4
Requirements Analysis




```
package A is
  procedure M...
end A;
------------------------
package body A is
  package BB is
    procedure...
  end BB;
  package CC is
    procedure...
  end CC;
  package body BB is separate;
  package body CC is separate;
        .
        .
        .
  procedure AA is
  begin
  end AA;
end A;
```

Program Design Language (PDL)

```
package Landing_Gear is
        .
        .
procedure Update_Landing_Gear_System
      (Inputs: in Landing_Gear_Inputs
       Outputs: out Landing_Gear_Inputs;)
        .
        .
end Landing_Gear;
```

Figure 5
Design Analysis

1) The methodology should not encourage the structure of the software to become a requirement. These types of approaches usually try to force a method by which detailed requirements are produced because no other method will quantify the requirements. This methodology also results from a design perspective in which the personnel involved do not understand requirements development and arbitration with the customer. Point one: Many designers feel they have inadequate requirements. It is inherent in their chosen field that they rely on some bounding of their problem. The more detailed the requirements, the easier the task becomes. The easier one can make a design task, the better, but putting detailed design abstractions in a baselined document that specifies requirements is a problem. For example, specifying a boolean internal within a CSCI prohibits the design from choosing to implement the interface in another manner (consider the strong typing of Ada). Many argue that the requirements documentation is never baselined to Critical Design Review (CDR), so who cares if design is a part of the requirements? Why is this the case? Many programs today just get off on the wrong path and stay that way. Contractors must start using the review process for what it is to be used for! Example: The purpose of a Mil-Std-1521 System Requirements Review (SRR) is not to present design, but to arbitrate requirements. All system level requirements must be identified and agreed upon before software requirements can be generated. This brings us to warning 2.

2) The methodology should not provide requirements documentation that will cause a long laborious testing phase. If detailed design abstractions are represented in the requirements documents, how long would it take to functionally test and verify/validate the design vs. the requirements? Programs

would take twice as long to finish; Congress would have a field day!

3) One of the programmatic problems which is consistently identified in large software development projects is the lack of adequate time spent in requirements analysis. Some object-oriented methods encourage more time in requirements analysis. This may be an artificial need. Pushing design activities into the requirements analysis phase expands the scope of the phase, thus redefining the phase.

Object-oriented approaches should encourage design prototypes during requirements analysis in order to address CSCI implementation problems as soon as possible in the development phase. Mil-Std-1803 encourages the use of prototyping for uncovering design implications. Building Ada code in requirements analysis or even in preliminary design often causes many to believe a premature coding phase has begun. In actuality, the use of the Ada language to prototype interfaces, language features, and common utilities helps provide the assurance that the requirements can be implemented. In addition, the Ada package specifications developed during this prototype provides a contract between subcontractors. The enigma of premature coding doesn't exist anymore. There are far more advantages to using the capabilities of the Ada language and object-oriented approaches to prototype affirmation of requirements implementation than disadvantages. The advantages include: Confirmation of requirements implementation strategies, enforceable compiled contracts between organizations, "seamless" transition from requirements analysis to design, uncovering of design problems early, and building of common utilities to provide a foundation of reusable components. It is imperative that designers participate in the requirements analysis phase to contribute to the successful completion of that phase.

This paper has addressed training software development from an object-oriented perspective and an acquisition-through-the-prime requirement. Techniques have been presented for the identification, specification, and documentation of common objects. Common software objects developed early in the program will ease the training systems development. This paper has presented an object-oriented requirements analysis approach that is compatible with DoD-STD-2167A, and explains an approach for using capabilities of the Ada language to ensure a proper transition from requirements analysis to design.

About the Author

Mr. Hendrix is a senior specialist engineer with the Boeing Defense and Space Group - Huntsville. He has been involved in Ada software development for training systems and tactical hardware since 1986. Mr. Hendrix was employed by General Dynamics in support of the F-22 program. He is currently assigned to the Space Station Freedom Program software engineering organization for the Boeing Company.

SOFTWARE RELIABILITY MEASUREMENT ON THE B-2 AIRCREW TRAINING DEVICE (ATD)

Bruce R. Bedford, Senior Staff Engineer
CAE-Link Corporation
Binghamton, NY

ABSTRACT

As the developer of the B-2 ATD, CAE-Link was tasked with building a very complex, software intensive training device. The development of the software was directly influenced by many leading edge technologies and philosophies. The B-2 ATD was the first project at CAE-Link to use Ada, object-oriented design, DOD-STD-2167A, and modern software reliability techniques. Specifically, our customer requested that we address software reliability measurement issues with creativity and innovative techniques. We chose to use McCabe's Cyclomatic Complexity Metric and Musa's Basic Execution Time Model. This paper covers the history of the B-2 ATD startup with software reliability measurement, our research in the area, the plan that we employed, and our experiences as the plan was implemented. It also includes data gathered through the use of automated tools, as well as remaining planned activity. Complexity, testability, maintainability, mean time between failures, education, and practical application all impacted our first real experience with software reliability measurement and are included in this paper. As the size and complexity of the software within aircrew training devices continue to grow, we must strive to find methods to measure its reliability. This paper is presented, in the words of software reliability pioneer John Musa, "in order that others may stand on our shoulders, rather than our feet."

INTRODUCTION (THE NEED)

Software reliability measurement is a fast growing technology. An awareness of the impact of unreliable software continues to grow among both industry representatives as well as their government counterparts. Many RFPs are now requesting some degree of software reliability measurement. In some cases when the request is made, both the contractor and the government program office are unsure of exactly what is to be measured and why. Nonetheless, there is a recognition from both parties that this is not a topic to ignore.

In the past few years, the government and Department of Defense (DOD) have undertaken a number of efforts in this area. AFSCP 800-14 - Software Quality Indicators is a pamphlet describing measurable indicators to gain insight into the quality of the software products being developed, including software reliability. AFSCP 800-43 - Software Management Indicators provides indicators into the software development process. Both documents are to be used by acquisition managers and their industry partners, to be tailored for specific projects. Another example is R&M 2000, an Air Force initiative that includes software reliability and maintainability objectives into the year 2000.

In industry, various organizations have undertaken initiatives of their own. The Electronic Industries Association has a subcommitee of their Computer Resources Committee to deal directly with software reliability issues. IEEE has formed the Technical Subcommittee on Software Reliability Engineering. Their first newsletter appeared in the January 1991 issue of IEEE Software. Also from IEEE, the Dictionary of Measures To Produce Reliable Software (IEEE standard 982.1-1988) contains a set of measures and short descriptions on their use. Conferences and symposiums on the topic are beginning to spring up. An example is the International Symposium on Software Reliability Engineering held in May 1991 in Austin, Texas.

Many of these, and dozens of other initiatives, did not exist just a few years ago. The field appears to be exploding. As much of this work was starting, the B-2 ATD development contract was awarded to CAE-Link. From a software reliability standpoint, we started work with our customer using a blank sheet of paper. What follows is an account of what we put on that paper.

CONCERNS

From the start, there were a number of issues that affected the entire issue of software reliability measurement:

- The "It won't work" syndrome. This is standard for any new area and must be expected from some circles. However, since our customer had asked us to do SOMETHING inno-

vative and creative in this area, we focused on creating something small and workable.

- In this field there seem to be many opinions and few conclusions. We tried not to let this get in our way. We knew from the outset that there were many tentacles to this issue. Our job was to focus on those opinions/conclusions that could help us.
- "Paralysis by analysis". We did not have much experience in this field, but this was a great opportunity to get a jump start and begin moving. Doing something about a problem is usually better than doing nothing.
- Establish a company experience base. If nothing more, when the project is finished, we will have a foundation upon which to build.
- Both management and engineering understanding. In a field where the purpose is to MEASURE the software, software engineers could quickly get the idea that THEY were being measured, not the software. It is critical that the information collected and derived be used properly.

OUR APPROACH

At the beginning, we tried to keep in perspective that whatever we chose to do, the approach had to be both usable and cost-effective. Usability is especially important from an engineering standpoint. If we attempted to apply some metrics that general engineering would not embrace, it would just be an academic exercise. Obviously, cost is always a factor. But when dealing with a new field, you can quickly lose management support for your ideas if there is a high initial cost with unproven returns in the future.

We divided the problem into two parts. First, we would spend some time researching the field and then we would put together a plan. The research took us down many roads. Some were fruitful, others not. As we asked questions using the phrase "software reliability", we got answers ranging from "software doesn't break or wear out, so it must be 100% reliable" to "you can't apply hardware reliability analysis to software". It can have a number of definitions, depending on the background and circle of people defining the term. Therefore, we had to limit the scope of what we would attempt to do. We felt it was critical to start small and succeed before tackling larger issues.

This paper is about the two measures that we chose to use and why. First, during software development, we would apply McCabe's Cyclomatic Complexity Metric.[1] Later, during system testing, we would apply John Musa's Basic Execution Time Model.[2]

WHY MCCABE

The McCabe Cyclomatic Complexity Metric provides a relative number for a unit of software. It is based on the branching logic within that software unit. It can be used to measure the complexity of both Program Design Language (PDL) and units of code. Its main use would therefore be during the development phases of a project.

The immediate question that comes to mind is, "What does unit complexity have to do with reliability?". McCabe's belief is that by limiting the complexity to a reasonable and understandable level, the testability and maintainability of that unit improves. We then made the intuitive link that the reliability of that unit would also increase. After all, if you can increase the testability of a unit of software, you have increased the chances of verifying all inputs to that unit and understanding all possible outputs. It then follows that the unit is more reliable, regardless of how you wish to define "software reliability" for that unit.

Another reason we chose to use this measure is because of lines of code limits (or rather, a lack of them). Many simulation projects have an imposed limit for the number of lines of code per software unit. The B-2 ATD does not. However, we felt that if we created complexity limits, we would directly influence the testability of that unit. Instead of having an arbitrary lines-of-code size (e.g., 100), concentrating on the complexity would also be more flexible. Programmers would not have to be concerned about "cutting off" the unit because of its length, usually at the expense of increasing interfaces. Instead they would be more concerned with the construction of its logic and how hard it would be to unit test. The details of our complexity guidelines are presented in the next section.

The final consideration was the availability of tools to calculate the Cyclomatic Complexity. It can be computed by hand, but with thousands of units that would not be cost effective. At the time we were doing this analysis, the project was already committed to using an Ada development tool, ADADL (Ada-based Design And Documentation Language). McCabe's Cyclomatic Complexity Metric is reported as a natural byproduct when software is developed with ADADL. This fell in nicely with one of our original objectives: cost. This measure was going to be available, now all we had to do was wrap a usable process around it. Also, McCabe & Associates markets a tool, ACT (Analysis of Complexity

Tool), that provides the Cyclomatic Complexity and much more. Although our original plan did not call for its use, we did determine later that it would help us. More on that later.

MCCABE EXPERIENCE

The plan to use this complexity measure needed some guidelines. Table 1 depicts both the guidelines and goals that we developed for use on the B-2 ATD. It is important to know that these were purposely made flexible.

First the guidelines. McCabe recommends that software units be kept to a Cyclomatic Complexity of around 10. This seems to be a limit where units become harder to understand and ensuring complete unit testing can become a problem. However, we felt that there were a couple of overriding issues with this limit. First, we develop real-time software. It must be fast. We couldn't impose a complexity limit on software that may tend to force the execution speed to suffer. An example is some flight software that must be written exactly as the aircraft manufacturer has written it, with no modifications. Also, our object-oriented design approach had an impact. Since controller units would necessarily have logic to make system decisions and act on statuses, their complexities would be at the high end. The object units, doing mostly data manipulation work, would have complexities at the lower end. Therefore, we set up the guidelines to provide visibility into possible problem areas.

As noted in Table 1, if the complexity was between 11 and 20, we required a written justification by the developer and the supervisor's approval. Over 20 required one more level of approval. This visibility provided checkpoints that prevented later problems in testing. For example, if a particular system was not overly difficult, but a young programmer developed software that was too complex, management found out early enough to make changes before they were too expensive. On the other hand, if a manager approved a particular high complexity, he was aware that the the system might be hard to test and he could plan accordingly. Lastly, the guidelines improved communication. The Cyclomatic Complexity of each unit was presented at the code walkthroughs and resides in that system's Software Development File (SDF).

The goals in Table 1 are just that. They are not requirements. Before any PDL or code was written we put these goals in the plan in order to have a gauge to measure software on a large scale. They were arbitrary. The 80% goal of Ada units less than or equal to 10 was based on the reasoning that when design matures into code, the complexity of the logic naturally increases. If a unit has a PDL complexity near 10, then as it is expanded into code it probably will exceed 10.

Tables 2 and 3 contain actual data from the B-2 ATD. Table 2 contains department totals for Ada Cyclomatic Complexities. It's interesting to note that we far exceeded our goal of 80% for the code complexities 10 and under. But the number of units, in all departments, greater than 20 is higher than 1%. One explanation is that our original arbitrary goals weren't realistic. The goal for units greater than 20 might be better at 5%. It may be unreasonable to believe that a system with well over a million lines of Ada should have less than 1% of its compilation units over 20. Also, maybe the percent for 10 and under should be around 90% instead of 80%. It might make for a more testable, reliable system. However, as a first attempt, I believe we are doing very well and these numbers support the conclusion that the real-time software is of high quality.

Table 1 Guidelines and Goals

GUIDELINES:

PDL CYCLOMATIC COMPLEXITY	CODE CYCLOMATIC COMPLEXITY	STATUS
0 - 10	0 - 10	ACCEPTABLE
11 - 20	11 - 20	ACCEPTABLE WITH WRITTEN JUSTIFICATION AND ONE LEVEL OF MANAGEMENT APPROVAL
21 +	21+	ACCEPTABLE WITH WRITTEN JUSTIFICATION AND TWO LEVELS OF MANAGEMENT APPROVAL

GOALS:

PDL	CODE
90% OF PDL UNITS <= 10	80% OF ADA UNITS <= 10
	1% OF ADA UNITS > 20

A couple of notes about Table 3. Analyzing this sort of table can give a software manager some insights into his systems. For example, System J has only 71.4% of its units with a complexity of 10 or less. But no units are greater than 20. If the two that are between 11 and 20 are closer to 11, the system should be quite testable and maintainable. On the other hand, System Y has about the same percentage of units 10 and under - 72.7%. But it has two units that are greater than 20. If they are significantly greater, and the design of the system is such that these two units are global in nature, System Y may indeed have some reliability problems. This type of analysis can be the catalyst to asking questions that will save both time and money in the test phases.

As we started into the regular routine of using ADADL and getting its complexity reports, we began to realize that it might be helpful if we knew more about the complexities than just the number. We contacted McCabe & Associates and obtained a video tape of a presentation describing methods of complexity analysis and the complexity impact on testing. It also demonstrated the Analysis of Complexity Tool (ACT). After we showed the tape to key lead engineers and their managers, they requested that it be shown as an educational and training aid to our software developers. In turn, many of them requested that management purchase ACT to help them in their testing. We did so, and it was quite helpful. Some engineers reported that the test path generator saved them hours of work. The flowgraphs were also put to good use. Engineers reported spotting dead or redundant code in their software and were able to more efficiently structure their code. One note on the tool itself. The Ada version of ACT had a small bug. There was some difficulty finding the problems, but the bug only affected a very small number of units. The impact was minimal.

MCCABE ISSUES

The first issue has to do with education. Our experience showed that it was not really important to get caught up in whether a unit's complexity was just under 10 or just over. At the beginning, a few engineers took the guidelines and tried to ensure that they had nothing over 10. They didn't want to have any perception that they were creating "bad" software. On the other hand, we had to draw the line in the sand somewhere. It was very important to get across the idea that THEY were not being measured. We had training sessions with all of the departments to explain the goals and guidelines, and to personally answer questions.

The next issue is the complexity derived from using the CASE statement. The way McCabe defines the Cyclomatic Complexity means that using a CASE construct will drive the complexity up rapidly (10 CASE options = complexity of 10). Without attempting to argue the validity of methodology, we simply gave an exception to the guidelines for the use of the CASE. However, it was not carte blanche. If a unit had a complexity of 30, but it contained a CASE statement with only 5 options, we still required the justification and management approval because that left the rest of the unit with 25.

The last issue is "trend vs. actual". Some systems lean toward higher complexities. I believe those systems with many units of moderately high complexities may have more problems than a system with a single unit of excessive complexity. Programmers tend to work in patterns with a certain style. If one's "style" is given to routine high complexity, it's easy to create a monster system that is not understandable, testable, or maintainable. A system that exhibits,a trend toward low complexity units, but has a single anomaly of high complexity, usually has a good reason. We had a system that

Table 2 McCabe Code Complexities by Department

NAME	TOTAL UNITS	<=10		11-20	>20	
DEPARTMENT TOTALS						
DEPARTMENT 1	1301	1213	93.2%	67	21	1.6%
DEPARTMENT 2	1746	1511	86.5%	58	77	4.4%
DEPARTMENT 3	879	782	89.0%	68	29	3.3%
DEPARTMENT 4	873	798	91.4%	45	30	3.4%
PROGRAM GOALS			>80%			<1%

Table 3 Department 3 Code Complexities by System

NAME	TOTAL UNITS	<=10		11-20	>20	
DEPARTMENT 3						
SYSTEM A	20	20	100.0%	0	0	0.0%
SYSTEM B	104	93	59.4%	8	3	2.9%
SYSTEM C	35	35	100.0%	0	0	0.0%
SYSTEM D	49	48	98.0%	1	0	0.0%
SYSTEM E	46	45	97.8%	0	1	2.2%
SYSTEM F	11	11	100.0%	0	0	0.0%
SYSTEM G	33	31	93.9%	2	0	0.0%
SYSTEM H	62	52	83.9%	6	4	6.5%
SYSTEM I	8	6	75.0%	2	0	0.0%
SYSTEM J	7	5	71.4%	2	0	0.0%
SYSTEM K	50	42	84.0%	8	0	0.0%
SYSTEM L	10	10	100.0%	0	0	0.0%
SYSTEM M	42	37	88.1%	1	4	9.5%
SYSTEM N	7	5	71.4%	2	0	0.0%
SYSTEM O	29	24	82.8%	3	2	6.9%
SYSTEM P	6	5	83.3%	1	0	0.0%
SYSTEM Q	10	10	100.0%	0	0	0.0%
SYSTEM R	14	14	100.0%	0	0	0.0%
SYSTEM S	43	34	79.1%	6	3	7.0%
SYSTEM T	117	107	81.5%	8	2	1.7%
SYSTEM U	21	18	85.7%	2	1	4.8%
SYSTEM V	31	26	83.9%	4	1	3.2%
SYSTEM W	24	13	54.2%	7	4	16.7%
SYSTEM X	32	29	90.6%	1	2	6.3%
SYSTEM Y	11	8	72.7%	1	2	18.2%
SYSTEM Z	8	7	87.5%	1	0	0.0%
SYSTEM AA	22	20	90.9%	2	0	0.0%
SYSTEM BB	27	27	100.0%	0	0	0.0%
DEPARTMENT 3	879	782	89.0%	68	29	3.3%
PROGRAM GOALS			>80%			<1%

contained a unit which had a couple dozen sequential, single branch IF statements. Each IF checked a different hardware status and then set a flag. Its complexity was well over 20. But it was under an execution time constraint. It was logical, understandable, and testable. This illustrates that there are exceptions to every rule, but that a negative trend can point out a significant problem.

WHY MUSA

When our research in this field brought us to software reliability models, we found a number of academic endeavors, but very few had been applied in industry settings. John Musa of AT&T Bell Laboratories has been involved with this field for almost twenty years. He has written numerous research papers and is an acknowledged leader in the field. But there were a couple of issues that forced us to look at his work a little more closely.

First, Musa coauthored a textbook titled SOFTWARE RELIABILITY: MEASUREMENT, PREDICTION, APPLICATION. At the time of its publication in 1987, this textbook essentially became the standard for the field. In our initial approach, we had the objectives of usability and cost-effectiveness. This book helped us toward the objective of usability. We did not wish to verify anyone's hypotheses or validate their algorithms to ensure that they were correct. Instead, we wanted to use a metric already developed. The book provided helpful hints in getting started and an approach that lent itself to moving along – getting on with it.

Second, we needed a software tool to perform model analysis. As we dug deeper into the subject, we contacted Musa. At the time, they had developed a public domain software tool to perform the derivations and calculations described in his book. It was not supported, but was still being used within AT&T. He offered to send it to us, with some porting instructions so that we could use it on our hardware suite. This tool met our cost-effectiveness objective. Since that time, AT&T has marketed a software reliability tool and it is available through their UNIX TOOLCHEST.

Musa defines software reliability as "... the probability of failure-free operation of a computer pro-

gram for a specified time in a specified environment". Musa's Basic Execution Time Model is an estimation and prediction model that is to be used during the system test phase of software development. It provides estimations and predictions (with confidence limits) for numerous items, including present MTBF, total failures, number of failures to find in order to meet an MTBF objective, and the time to find those failures. The model is based on the ability to collect the CPU time between software failures. CPU time is the "specified time" in the definition. He uses CPU execution time because it is a gauge as to how long the software is in use. Calendar time (or wall clock time) is used in hardware reliability because the failure rate of the hardware is a function of long it has been in use. In software, components don't wear out. However, the more the software is executed (i.e., tested), the more bugs are found and fixed, and the software becomes more reliable. One of the model's main features is to provide insight into the status of the software during the system test phase. In system testing, sometimes it is difficult to know when you're done. Musa's model is an attempt to estimate the present software MTBF of the system (the average time between software failures) and then predict how long it will take to meet an MTBF objective (when you will be done).

MUSA EXPERIENCE

Our experience thus far with the use of this model has been limited to preparing a foundation to use it. We are still a number of months away from gathering actual data and analyzing reports. However, there are already a few experiences that would be helpful to anyone entering this arena for the first time.

The first experience is that of porting the software tool. The public domain model was written to run on an IBM and we ported it to a VAX. The total time was not excessive (a couple of weeks). It is written in FORTRAN and the software uses two data files as input. One is the list of CPU time intervals between failures. This data is to be collected during testing. The second contains a series of parameters that the model uses to make predictions. Much of this information will be estimated at the beginning and then modified as you gain experience during the system test phase.

Another experience worth noting is that of education. The notion of measuring the reliability of software is quite foreign to most engineers with a reliability and maintainability background, software developers, software managers, as well as the software customer. Therefore, education plays an important role. We have purposely not "promised the world" with the use of Musa's Model. The approach has been one of "one step at a time". We have an understanding that this is new to most of us, but it seems to hold some promise. We want to use it as another piece of information during system testing, not the DECIDING piece of information. If we can provide another sanity check during a time when all eyes are on both the quality of the system and the delivery schedule, we will have provided a valuable service.

We have tested the process of gathering information, formatting the data, and producing a report. We have worked out a few kinks, such as ensuring that record formats are correct when we format the data gathered from the target machine (Concurrent) to processing it off-line (VAX). However, this has only been done in the early stages of Hardware–Software Integration (HSI). Our plan is to run through the process during the dry run of the Development Test Procedures (DTP). We then should be ready to use the model when the DTP is run formally.

MUSA ISSUES

The area of data collection is key. In real-time, the failures that we want to capture fall into two categories. The first is the type of failure that causes the applications software to abort. This type of error is handled through our real-time executive software and is reported to both the terminal and an error file. We worked with our real-time executive designer to ensure that the proper information was to be recorded into this error file. Once that end of data collection was satisfied, we then developed a formatter routine to extract the applicable data from the error file and modify it to be acceptable as input into the Musa model tool.

The second type of failure cannot be handled quite so directly. These types of failures are those in which the software does not abort, but instead executes to the end of its logic but the logic is incorrect. An example would be if a routine incorrectly calculates the aircraft fuel consumption, and then passes it along as a parameter to other software routines. This type of failure is usually found through analysis of results during system testing, but would obviously not cause an entry into an error recording file that handles run–time aborts. This type of a failure also needs an entry into the CPU intervals file. Musa suggests that you approximate, as close as possible, the time at which the failure occurred during testing, then manually enter the failure as a line item into the CPU intervals file. The premise is that recording an "approximate" time of the failure re-

sults in better accuracy from the model than if the failure was not included at all.

Another issue is test coverage. The model is going to process data gathered during testing. It is going to make estimations and predictions based on that data. If the data is representative of tests that cover the full range of input values, the model should be expected to be reasonably accurate. However, if the test coverage is lacking, the model may provide unrealistic information. It may report that the present MTBF of the software being tested is at an acceptable level. But if there is an area of software functions that are not tested thoroughly, the reported present MTBF would be misleading. We are handling this issue by gathering data during the formal running of each department's DTP. We feel that coverage provided by these tests at that level should be adequate.

The last issue is similar to an issue with using the McCabe Cyclomatic Complexity measure: trends vs. actuals. It is our plan to use the information provided by Musa's Model as a trend gauge. We feel, particularly at this stage of our experience, that to put too much faith in the results provided by a particular data set would not be in our best interest. On the other hand, evaluating a series of estimations and predictions over a period of time may provide valuable information about the status of our system testing.

FUTURE WORK

We are just beginning work in another area – producing reports based on the change activity of released software. We are presently working on the details of gathering the information and formatting these reports. The major thrust is to gain visibility into the amount of change activity taking place with particular software units and their systems. Initially, there will be a basic report to list the software units and the corresponding number of changes that a unit has had since it was placed under Configuration Control. A later report will plot the number of software units changed over time. This effort was not part of our original plan. Working with our customer, under the heading of software reliability, this task was initiated to gain insight into the maturity of our software through the change activity process.

CONCLUSIONS

This entire effort has provided a company experience base upon which we can build. The B-2 project has benefited from the use of the McCabe Cyclomatic Complexity measure in a number of ways. It can be a factor when evaluating bids for Engineering Change Proposals. It will have an impact on reusing Ada software and will influence the building of an Ada reuse library. And it certainly had a positive effect on the testing, maintainability, and understandability of much of our software. The possible benefits from the Musa Basic Execution Time Model are still in the future, but we are positioned well. As a minimum, the attempt to use the model has forced many questions to be asked and has allowed us to look at our software from a different vantage point. All of this work has also helped us to understand the enormous impact that reliable software has on our training systems. Just as the aircraft that we model, our training systems are becoming more and more software intensive. We therefore must continue to look for innovative and creative methods to measure the reliability of that software.

REFERENCES

1. McCabe, Thomas J. "Structured Testing: A Software Testing Methodology Using the Cyclomatic Complexity Metric", National Bureau of Standards Special Publication 500-99, 1982.

2. Musa, John D., Iannino, Anthony, Okumoto, Kazuhira. SOFTWARE RELIABILITY: MEASUREMENT, PREDICTION, APPLICATION, McGraw-Hill, Inc., 1987.

ABOUT THE AUTHOR

Bruce R. Bedford is a Senior Staff Engineer at CAE-Link Corporation, Binghamton, N.Y. Operations. Mr. Bedford's involvement with software reliability began in 1987. He has been with Link since 1983 and has held various positions within the company. Prior to joining Link, he was an Air Force officer stationed at Langley AFB., where he was a software developer on the Computer Assisted Force Management System (CAFMS). Mr. Bedford received his BS in Mathematics from Waynesburg College in 1974.

SOFTWARE METRICS, ADA, AND THE B-2 ATD

Paul E. McMahon, Staff Scientist
Dennis W. Meehl, Section Head
CAE-Link Corporation
Binghamton, New York

ABSTRACT

Many believe the greatest benefit of Ada is that it encourages software engineers to explore new design approaches leading to higher quality software. However, Ada's primary goal is to reduce the life cycle cost of software. Furthermore, the relationship between cost and modern software techniques is not always evident. This paper addresses the cost of Ada software. How long does it take an engineer to develop software when using Ada and modern software engineering techniques? How much computational capacity does Ada require? This paper provides answers to these questions based on data from the B-2 Aircrew Training Device (ATD). Lines of code, development time, and computational resources are provided for selected B-2 ATD software systems. Key contributing factors include the cost of training engineers in modern software techniques and the impact caused by developing and using more modern software tools. This paper identifies key factors found on the B-2 ATD to be influential in affecting today's software cost and explains what we are doing to reduce this impact in the future.

INTRODUCTION

The B-2 ATD was the first major training device at Link to use Ada. With over 1.7 million lines of code, it was also one of the most complex. Factors contributing to software cost on a first Ada project include on-the-job training in new design techniques and costs associated with new tools. The impact of these factors is expected to decrease on future Ada projects.

There are no simple answers to software metrics with Ada. Simple formulas fail to consider many of the possibilities. Ada metrics can be sensitive to design techniques and compiler implementations. These situations must be identified and managed. This paper is intended to provide information that can help in estimating, and also understanding, software costs with Ada.

Ten software components (CSCs in DOD-STD-2167A terminology) were selected for this study. Eight real-time and two non-real-time systems were investigated. The real-time systems were selected from the disciplines of aerodynamics and avionics. For the systems selected, the design engineer's experience in simulation ranged from 2 to 25 years. Ada experience ranged from the first to the fourth Ada assignment. The following data were obtained:

- Development Hours - Development hours were derived from our Management Control and Information System (MCIS). MCIS is a performance measurement system and is a validated Cost/Schedule Control System (C/SCS).
- Lines of Code - Lines of code were measured by a source code scanning tool. Both "carriage return" lines of code and "semicolon" lines are reported.
- Bytes of Memory - Memory requirements were derived from computer vendor load maps and measured stack space needs for program execution.
- Execution Time - Execution time was measured using a microsecond clock on the real-time target system.

HISTORY OF PROJECT

Resource Estimation

Initial computational resource estimates on the B-2 ATD were based on a Fortran to Ada translated benchmark known as Mainflt. Translating Mainflt resulted in Ada code employing primarily integer, float, and boolean data types.

This benchmark required 40 bytes per line of code and 1.25 times the equivalent Fortran execution time. Additional resources for new design techniques with Ada were unknown at the start of the project. The cost of new design techniques with Ada is discussed later in the paper.

Development Environment

The B-2 ATD software was developed using one of the most mature Ada development environments available today. After test, the software is

transferred through a local area network to the target system, where it is compiled again and linked. The target computer has its own distinct operating system and Ada compilation environment. A single processor on the target system provides approximately 6 VUPS (VAX units of processing speed) of computing power. The B-2 Weapon System Trainer (WST) requires approximately 20 processors, or 120 VUPS. The target provides a real-time non-virtual operating environment with limits, such as memory, that do not exist in the development environment.

Our experience indicates that the resource demands of Ada compilers may differ. It is not recommended that the numbers presented in this paper be applied to estimates for other Ada environments. This information should be viewed as trend data only. Benchmarking one's chosen compiler and target hardware is necessary to arrive at accurate resource estimates.

SOFTWARE STRUCTURE MODEL

Many simulation design issues are common, providing an opportunity for reuse. One vehicle that aids us in applying reuse at Link is our software structure model developed specifically for use with Ada. This model has been developed and coordinated with members of the Software Engineering Institute (SEI) staff.

In this section a brief description of the model and real-time environment is provided. This subject is discussed here because understanding the software structure is helpful in understanding some of the new cost trends with Ada. The structure model can also be used in reducing software costs.

Real-Time Environment

The B-2 simulation software runs in a tightly-coupled parallel-processor environment. Separate processors communicate through a global memory system. Application software does not directly access global memory except for time-critical applications. Communication through global memory, common file I/O services, and executive control are provided by automatically generated software.

Interface Management

During the design phase, global interfaces are defined through a data base referred to as the Interface Management Data Base (IMDB). An off-line processor uses the IMDB to generate GLOBAL DATA PACKAGES and IMPORT and EXPORT PROCEDURES. These procedures move the data in real time to and from the global packages at rates specified in a control file. Imports are moved to local IMPORT PACKAGES and exports are moved from DECLARATION PACKAGES. The automatically generated IMPORT and EXPORT procedures are referred to as CONNECTION MANAGERS.

Application Software

CONTROL MANAGERS are called at rates specified in a control file by an automatically generated EXECUTIVE. A CONTROL MANAGER is the top-level user procedure and usually controls software the equivalent of a DOD-STD-2167A CSC. However, a single control manager may control multiple CSCs, or a single CSC may have multiple control managers.

"Objects", in an object-oriented design (OOD) sense, are defined in OBJECT DEFINITION PACKAGES. These packages define Ada data types and Ada procedures that operate on these types. The CONTROL MANAGERS invoke these "objects", passing data from the IMPORT PACKAGES and DECLARATION PACKAGES. OBJECT DEFINITION PACKAGES may contain only types or types and procedures together.

Real-time application designers develop CONTROL MANAGERS, DECLARATION PACKAGES, IMPORT PACKAGES and DEFINITION PACKAGES. The EXECUTIVE, IMPORT and EXPORT CONNECTION MANAGERS, and GLOBAL DATA PACKAGES are automatically generated.

Real-time file I/O services required by the application code are created, using Ada's generic capability, based on the application types. Direct I/O is provided to application software for both local and remote file access. Figure 1 is a diagram of the Structure Model Components.

DEFINITIONS

Closure

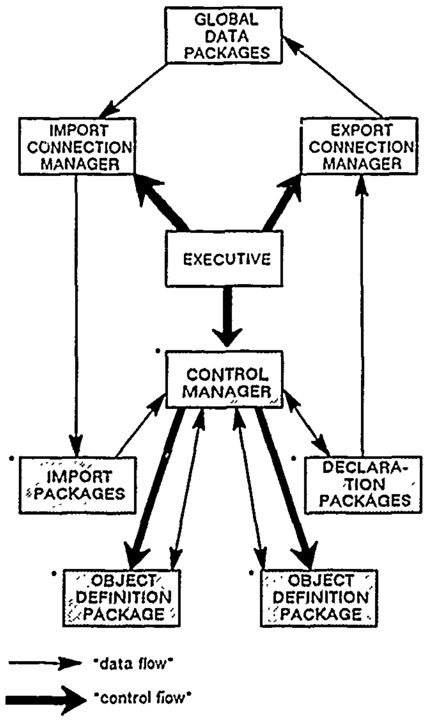
Closure consists of all the Ada units required in the library for a given unit to compile (compilation closure) or link (execution closure).

Design Phase

The design phase includes the development of the CONTROL MANAGER specifications, DEFINITION PACKAGE specifications, DECLARATION PACKAGES, and IMPORT PACKAGES.

Code and Test Phase

In the code and test phase the Ada bodies are completed for the CONTROL MANAGERS and the DEFINITION PACKAGES. Simulation algorithms reside in the bodies.



*Shaded areas are components developed by engineers.

Figure 1 Structure Model Components

Ada Lines of Code

In this study we report both the number of semicolons and the number of lines with carriage returns containing Ada compilable statements. Lines of code generated automatically by off-line processors and generic instantiations are reported separately and are not used in productivity calculations since engineers do not manually generate this software.

Compiler-Generated Code

Compiler-Generated Code consists of all executable instructions generated by the compiler. This includes elaboration code and user code.

User Code

User code consists of all compiler-generated code produced from Ada statements found after the BEGIN in procedure and function bodies. The code that carries out the simulation algorithms is user code.

Elaboration Code

Elaboration code consists of all compiler-generated code used solely to carry out MIL-STD-1815A elaboration rules. Elaboration code can be thought of as set-up code. It is only run in preparation for executing user code.

Static Data

Static data includes all Ada variables allocated to dedicated memory locations. Static data remain fixed in size and location throughout the simulation exercise.

Stack

The stack has two parts. First, the stack is used to elaborate packages. This only occurs once prior to the start of simulation. Secondly, the stack consists of temporary data used during simulation. This temporary data on the stack does not retain its value between program calls, does not remain fixed in location between program calls, and may not be fixed in size.

DATA ANALYSIS

For each of the systems analyzed, Table 1 provides numbers of Ada units, lines of code, and memory requirements. Table 2 provides experience levels of the software engineers assigned to develop these systems. Development time is discussed later in the paper.

This data indicates that traditional methods used to estimate computational resources may fall short with Ada. This is because there are new and influential factors with Ada. Such factors include the use of composite types (records and arrays), elaboration code, overhead for packaging system services, stacks, and the use of generics. These five factors are discussed in the following paragraphs.

Composite Types in Ada

The resources required for a single Ada statement can vary dramatically as a result of Ada's composite type capabilities. Consider the data for Forces_And_Moments, Aero_Coefficients, and Nav_Geography in Table 1. These systems average 66, 41, and 39 bytes per semicolon per declaration package. The data in these systems consists primarily of "small" records each containing 5-10 scalar components. On the other hand, the Mass_Storage_Unit CSC averages 1930 bytes per semicolon per declaration packages. This is due to a single declaration requiring over 150 kilobytes.

In many cases with Ada, we are seeing complexity moving out of the code and into the data. Data structures can become complex rapidly. Evidence of this fact is found in the size of the declaration packages.

Table 1 B-2 ATD Lines of Code and Memory Requirements

Structure (Note 1)	No. of Ada Units	No. of Ada Lines <CRs>	No. of ;s	Constants	Static Data	Compiler Generated Code	Total Bytes	Bytes Per ;	Avg. ; Per Unit
Real-Time Systems									
Forces and Moments									
Defns	5	388	209	140	544	196	880	4	42
Declar	2	108	51	40	1488	1848	3376	66	26
Bodies	5	449	216	20	80	6260	6320	29	43
Exp Imp	2	586	270	112	48	2272	2432	9	135
Total	12	945	476	200	2112	8264	10576	22	40
Mass Storage Unit									
Defns	6	1049	795	1752	368	24	2144	3	133
Declar	4	136	85	2184	161504	424	164112	1930	21
Bodies	5	4640	2321	3400	4848	70168	78416	34	464
Exp Imp	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Rtio	3	36	21	28	10736	164	10928	520	7
Total	15	5825	3201	7336	166720	70616	244672	76	213
Weight and Balance									
Defns	2	193	141	136	160	184	480	3	71
Declar	2	374	130	116	4200	6844	18160	140	65
Bodies	10	744	387	136	160	10360	10656	28	39
Exp Imp	2	2058	1059	636	32	6116	6787	6	530
Total	14	1311	658	388	11520	17388	29296	45	47
Aero Coefficients									
Defns	7	423	329	144	1872	2312	4328	13	47
Declar	2	60	57	24	720	1608	2352	41	29
Bodies	19	2494	1065	196	1328	30488	32012	30	56
Exp Imp	2	575	286	252	48	6100	6400	22	143
Total*	28	2977	1451	364	3920	34408	38692	27	52
Nav Geography									
Defns	2	345	297	48	96	16	160	1	74
Declar	2	183	81	760	1408	952	3120	39	41
Bodies	2	2198	856	1244	400	16468	18112	21	428
Exp Imp	2	428	215	16	48	2540	2604	12	108
Total	6	2726	1234	2052	1904	17436	21396	17	154
Motion									
Defns	10	713	453	992	352	848	2192	5	45
Declar	2	181	42	24	2432	216	2672	64	21
Bodies	10	1402	703	204	224	17700	18128	26	70
Exp Imp	2	428	250	224	80	4960	5264	21	125
Gen Ins	11	N/A	N/A	92	192	7620	7904	N/A	N/A
Total	22	2296	1198	1220	3008	18764	22992	19	55

Note 1:
- Exp Imp means EXPORT/IMPORT PROCEDURES. See Software Structure Model.
- Rtio means Real-Time I/O Auto-Generated Software
- GenIns means Generic Instantiation
- Defns means DEFINITION PACKAGES. See Software Structure Model.
- Declar means DECLARATION and IMPORT PACKAGES. See Software Structure Model.

Note 2:
- Totals do not include automatically generated software.
- This includes Export/Import software, Real-Time I/O, and Generic instantiations.

Table 1 B-2 ATD Lines of Code and Memory Requirements (Cont'd)

Structure (Note 1)	No. of Ada Units	No. of Ada Lines <CRs>	No. of ;s	Constants	Static Data	Compiler Generated Code	Total Bytes	Bytes Per ;	Avg. ; Per Unit
Motion Supplement									
Defns	18	810	443	1036	1792	68	2896	7	25
Declar	2	269	99	2356	3376	460	6192	63	50
Bodies	16	1274	665	204	816	15236	16256	25	42
Exp Imp	2	1103	424	84	48	4796	4828	12	212
Gen Ins	14	N/A	N/A	324	292	6312	6928	N/A	N/A
Total	36	2353	1207	3596	5984	15764	25344	21	34
Seat Shaker									
Defns	9	278	212	268	240	308	808	4	24
Declar	2	117	47	140	336	196	672	14	24
Bodies	7	497	244	108	160	7684	7952	33	35
Exp Imp	2	298	190	120	64	3912	4096	22	95
Gen Ins	2	N/A	N/A	16	32	688	736	N/A	N/A
Total	18	892	503	516	736	8188	9432	19	28
LFI Data (Sample)									
1 Var	N/A	32	16	92	176	468	736	46	N/A
2 Var	N/A	48	16	364	672	564	1600	100	N/A
3 Var	N/A	855	68	11160	21984	3608	36752	540	N/A
Real-Time Totals									
Defns	59		2879				13888	5	
Declar	18		592				200656	339	
Bodies	74		6457				187852	29	
Exp/Imp	14		2694				32508	12	
Gen Ins	27		N/A				15568	N/A	
Non-Real-Time Systems									
Master Bootstrap									
Defns	9	461	292	1112	12480	1732	15324	52	32
Declar	6	30	30	444	12544	1872	14860	495	5
Bodies	40	2290	1343	18024	1148	73700	92870	69	34
Total	55	2781	1665	19580	26172	77304	123056	74	30
Slave Bootstrap									
Defns	2	12	8	24	48	8	80	10	4
Declar	1	7	5	12	16	4	32	6	5
Bodies	22	1154	660	2996	608	25352	28956	44	30
Total	25	1173	673	3032	672	25364	29068	43	27

Note 1:
- Exp Imp means EXPORT/IMPORT PROCEDURES. See Software Structure Model.
- Rtio means Real-Time I/O Auto-Generated Software
- GenIns means Generic Instantiation
- Defns means DEFINITION PACKAGES. See Software Structure Model.
- Declar means DECLARATION and IMPORT PACKAGES. See Software Structure Model.

Note 2:
- Totals do not include automatically generated software.
- This includes Export/Import software, Real-Time I/O, and Generic instantiations.

Table 2 Personnel Experience

System	Engineer	Years of Experience in Simulation	Years of Prior Ada Experience
Forces and Moments, Aero Coefficients, Weight and Balance	A	25	0
Mass Storage Unit	B	2	3
Slave Bootstrap, Master Bootstrap	C	1	1
Nav Geography	D	15	0
Motion, Seat Shaker, Supplemental Motion	E	9	0

Declaration packages tend to require significantly more memory per semicolon than other structural components. On average, the B-2 data indicates that our estimate of 40 bytes per semicolon, based on our Mainfilt benchmark, is valid for definition packages and bodies. For declaration packages this estimate may be off by a factor as great as 10 or more. The average number of bytes per semicolon of all measured declaration packages was 375 (339 for real-time alone).

However, by not including the large data structure within the Mass_Storage_Unit, the average number of bytes per semicolon for declaration packages drops to 77. This analysis may indicate that large data structures within the application software should be accounted for together with other large data structures such as global/hardware interfaces. As a result, one can use a single "bytes per semicolon" estimate treating all structural components as code. This simplifies the resource estimation process for the application designer.

Elaboration Code

In the data analyzed, only about half of the memory required for declaration packages was allocated for user data. The other half was elaboration code. Any compiler-generated code in a declaration or definition package is elaboration code. User code resides only in bodies.

Ada compilers today can generate significant amounts of code in carrying out the elaboration rules of Ada. Elaboration code can be particularly costly in initializing data of a composite type. As an example, compilers may generate elaboration code to initialize records and arrays even when the initial values are known at compilation time. Compiler enhancements in this area can result in significant memory savings.

OVERHEAD COSTS

The memory demands of Ada extend beyond the application software itself. Memory required for stacks, generic instantiations, and "closure" units adds to the total resource picture.

Closure

All units in another unit's closure do not necessarily need to be loaded into memory at execution time. A package specification that contains only type information may be needed only during compilation. However, when "withed" units contain information that may change at run time or cannot be determined at compilation time, this unit may need to be loaded at run time.

When multiple procedures and/or functions are placed within a single Ada package, all of the software or only those procedures or functions actually referenced may require memory. These memory issues are dependent on compiler vendor implementations and can result in different memory demands between compilers.

An Example of Overhead Costs

On the B-2 ATD, Bootstrap is a set of interactive OS processes providing a menu-driven capability to initiate various simulator functions, such as simulator loading. Slave_Bootstrap provides control for a single OS environment and Master_Bootstrap provides common control over all Slave_Bootstraps. The total amount of memory required by the Slave_Bootstrap application software is 95 kilobytes. This includes 65 KB of software reused from Master_Bootstrap. However, the OS services required for such functions as loading and starting tasks require another 120 kilobytes. The package text_io requires 200 kilobytes. The stack requires 110 kilobytes and closure units add 35 kilobytes. As a result, the total task size of Slave_Bootstrap is 465 kilobytes. Master_Bootstrap requires 400 kilobytes.

Stacks

Most simulation processes on the B-2 ATD require approximately a 500 KB stack. Stack demands, however, are highly dependent on user software characteristics. It is not unusual for some applications to require stacks as large as 2 MB or larger. This includes the stack space required to initially elaborate packages as well as the space needed to execute the simulation programs. Determining worst-case stack requirements may demand special tools. We have used a specially developed vendor tool reporting worst-case stack needs to assist us in managing this resource.

Generics

Generic instantiations result in complete software units from a single Ada statement. Dependent on the compiler, these units may require computational resources similar to manually generated units.

Most of the systems analyzed do not use generics. Weight_and_Balance, Aero_Coefficients, Nav_Geography, and Forces_and_Moments do not utilize generics. It is not uncommon for first and second Ada systems to make little use of generics. Generics tend to be used by more experienced Ada engineers. However, when generics are employed, their resource demands can be substantial. Large

quantities of code requiring significant resources can be generated rapidly when using generics. Our Motion and Motion Supplemental CSCs utilize generics (see Table 1).

Unconstrained Types

Unconstrained types in Ada provide simulation software engineers with a powerful new capability for managing data. The resource demands of this new feature, however, may be more costly than expected.

An Example

On the B-2 ATD project, an off-line processor called the LFI (Linear Function Interpolation) Compiler is used to transform aircraft data sets into an Ada-compilable format for use in the training device. The data is interpolated in real time, frequently at high rates. Since each data set may vary in size, the use of an unconstrained Ada type was chosen.

We found with our real-time compiler that objects of an unconstrained record type required more computational time and memory than expected. In certain cases the maximum size of an object rather than the actual size was allocated both on the stack and in memory. This maximum was almost 2 MB. Furthermore, additional computational time was required since these objects were being passed on the stack as parameters to an interpolation routine. Objects of a constrained record type are passed as parameters more efficiently.

Limits and Automatically Generated (Auto-Gen) Software

We also found that some of the larger LFI data sets caused a constant table limit to be exceeded within the compiler and used excessive stack space during elaboration which was never recovered.

The LFI data sets are not the only Ada software automatically generated (Auto-Gen) on the B-2. Import and Export Procedures are automatically generated from the Interface Management Data Base (IMDB). Real-time disk I/O software employs generics. The Executive software is partially generic and partially automatically generated.

Automatically generated software can enhance productivity and reliability and is highly recommended. However, our experience indicates that real-time software employing unconstrained types and Auto-Gen software may have an increased risk of encountering target compiler limitations or inefficient use of resources. These conditions may not be evident when initially developing the software in a virtual non-real-time environment.

ADA CODE CHARACTERISTICS

Counting Ada Lines

Today there is not a single accepted standard for measuring lines of code in Ada. In this paper both lines with carriage returns and lines with semicolons are reported. We have seen approximately a 2 to 1 ratio between "carriage return" lines of Ada and semicolon lines. However, this relationship can vary depending on the particular Ada constructs employed and coding style. As an example, in the case of a 3-variable LFI (see Table 1) containing a large aggregate, this ratio is more than 10 to 1. Nevertheless, our experience indicates that managing size is best accomplished by focusing on terminating semicolons.

Comparing Ada to Other Languages

Our experience indicates that Ada may require more lines of code than languages such as Fortran. This is partially a result of design techniques and coding style, but it is also a result of the language itself.

Table 3 indicates that the ratio of code contained in the DEFINITION AND DECLARATION PACKAGES to the total manually generated code averages 41% for the real-time software analyzed. This may indicate that we can expect 40% more lines of code with Ada. This statistic also indicates that 40% of the Ada code generated occurs in the design phase.

Table 3 Percentage of Ada Generated During Design

System	Percentage of Design Code (DEFNS. DECLARS) to Total
Forces_And_Moments	54%
Mass_Storage_Unit	38%
Weight_And_Balance	41%
Aero_Coefficients	27%
Nav_Geography	31%
Motion	41%
Motion_Supplement	45%
Seat_Shaker	51%
Average	41%

In Fortran a preprocessor to compilation added interface declarations from a symbol dictionary. These declarations were not part of lines of code management. In Ada, these declarations are included in lines of code management.

EXECUTION TIME

Execution time can vary depending on loops and branch paths. Nevertheless, when code is primarily straight line and the data used is small records or scalars, estimates based on source line counts are possible.

Table 4 indicates a range of 0.5 microseconds to 4.1 microseconds per semicolon for the systems measured. Be advised that all of the instructions in these systems may not have been exercised during timing. However, this data indicates that Ada compilers today can generate code that meets stringent real-time simulation needs. In fact, we have found that managing real-time computational time is actually more of a simulation software design issue than a compiler issue.

Table 4 Real-Time Software Execution Time

System	Measured Execution Time*	Time* Per Semicolon**
Forces_And_Moments	240	1.1
Motion	320	0.5
Seat-Shaker	440	1.8
Nav_Geography	1482	1.7
Weight-And_Balance	1570	4.1

* Execution Time is reported in microseconds

** Manually generated bodies only are used for this calculation

The reader is cautioned against applying this data to code with varying design techniques and styles. Deep nesting of small procedures, the use of large composite data types, or the use of unconstrained types can dramatically alter timing results. For example, in a separate case study of a high-rate CSC on the B-2 ATD, 6-10 microseconds per semicolon was measured. Characteristics of this CSC included many array references, procedure calls, and loops.

DEVELOPMENT TIME

The ultimate success of Ada may rest with how favorably engineering productivity compares with previous generation languages. Table 5 provides the hours required to develop the eight real-time systems studied from the B-2 ATD. Productivity ranges from 0.7 to 4.0 Ada statements (semicolons) per hour. In all the cases analyzed productivity improved (frequently by 100% or more) moving from the design stage to code and test. This is believed to be due to two factors.

First, designing the definition packages requires considerably more thought than coding the bodies. Secondly, on a first and second Ada assignment, on-the-job training in Object Oriented Design (OOD), costs associated with new tools, and rework due to immature compilers all impact cost, particularly in the early design stages.

We have found that the OOD cost frequently includes a redesign of the first system and a refinement cost on second and third assignments. Despite this situation, our data indicates that developing software with Ada can be cost-competitive today. Once Ada experience has been gained, and a process and mature toolset put in place, further productivity gains can be expected. While individual results will vary, dramatic productivity gains can occur, as seen from our experience with the Mass_Storage_Unit.

RECOMMENDATIONS

Use Small Team Early

Ada's goal is to reduce the life cycle cost of software. New factors with Ada can increase complexity leading to higher, rather than lower, software maintenance costs. The use of small teams with focused goals can play a key role in managing this complexity early.

We recommend that a small team investigate the factors discussed in this paper as early in the project as possible. This activity must occur on the chosen real-time target system.

Our goal is to keep the software process simple. Software designers need clear and concise rules to achieve maintainable software. These rules must be based on the structure model, and on target-specific factors that can be learned only through prototyping. Rules for estimating resources can also be kept simple by managing large data structures as system data, allowing a common approach to all application structural components.

Table 5 Development Time

System	Design Time*	Code and Test Time*	; Per Hour Design	; Per Hour Bodies	; Per Hour Total
Mass_Storage_Unit	436	360	2.0	6.4	4.0
Nav_Geography	1200	700	0.3	1.2	0.7
Weight_And_Balance, Aero_Coefficients, Forces_And_Moments	1753	2140	0.5	0.8	0.7
Motion, Motion_Supplement, Seat_Shaker	1695	939	0.8	1.7	1.1

* Time reported in hours

Data types used for system interfaces must be established and controlled early. Typing strategies must consider both the value of modern software techniques and the "side effects" that may occur on one's chosen machine. The impact of data structures (such as structures using unconstrained and composite types) on stacks, elaboration code, and closure software must be known early and considered in establishing the rules.

Use Compiler Options When the Software Is Mature

We have found that once the real-time software is mature, significant computational resource savings can be gained through compilation options. While individual results may vary, we have seen execution time reductions of 30% when disabling run-time checks and a 25% savings in memory. By in-lining small units (less than 5 semicolons) another 10% reduction in execution time can be achieved, although in-lining does increase memory.

To gain these benefits it is important not to "design in" Ada's run-time checks. For example, one should not employ constraint exception processing to limit data. Otherwise, the software will not operate properly when the checks are disabled.

We recommend that these options be employed only toward the end of the project when the software is mature and the full impact is clear. Ada's run-time checks (e.g., range checking of interface data, ranging of indexes, etc.) are invaluable during test and in-lining can increase recompilation during development when the software needs to be modified frequently.

DISCIPLINE AND COMPUTER SYSTEMS SUPPORT

Although hardware/software integration (HSI) time is not included in this study, our experience on the B-2 ATD project indicates that HSI time is considerably shorter with Ada. While the time to build a load is longer, fewer loads are required to attain functionality. Although this is partially due to Ada itself, it may also be partially a result of the discipline the host-target environment brings to the software process.

Most, if not all, projects face schedule pressures, especially late in integration. Mounting pressure to integrate more software faster doesn't change with Ada. However, providing development tools in a different environment from the real-time target motivates engineers to test more thoroughly prior to software release. On the B-2 ATD we have found the software released for integration to be significantly more mature, leading to shorter integration schedules.

However, we have also found that the application engineers require more computer systems support in the integration stage. This is due to the fact that the development tools they have become familiar with may not be available in the target environment. Having spent most of their time in development, they are simply not as familiar with the target tools. As a result, we have found a need to provide more target computer systems support during integration than on traditional programs where development occurs on the real-time target machine.

CONCLUSIONS

Ada is complex and introduces new software measurement factors and trends. With Ada we are seeing more lines of code, but we are also seeing higher productivity rates. Complexity is moving out of the code and into the data. Data declarations are requiring factors of 10 or more times traditional memory requirements. Design time is increasing, but code and test time is decreasing. There are new costs associated with language features such as composite types, unconstrained types, and generics. There are also many new factors to consider in managing computational resources with Ada; stack space, closure software, system services, generics, and structural overhead all must be closely managed.

Today, compilers, tools, and Ada environments are rapidly maturing. The cost of computational hardware is continually decreasing. Clear and simple rules supporting modern software techniques can lead to increased productivity and reduced software costs with Ada. This is being realized today on the B-2 ATD project.

ABOUT THE AUTHORS

Paul E. McMahon is a Staff Scientist at the Binghamton Operations of CAE-Link Corporation. Mr. McMahon has been with Link since 1973 and has held various management and technical positions within the company. He has been involved with Ada development on the B-2 project since 1985. He received his BA in Mathematics from the University of Scranton in 1971 (Magna Cum Laude) and his MA in Mathematics from the State University of New York at Binghamton in 1973. Mr. McMahon has published numerous papers on Ada dating back to 1985. His most recent publications include a paper entitled "On the Fringe of Ada", presented at the 1989 NAECON Conference, a paper entitled "Lessons Learned on the Fringe of Ada", which was nominated for best paper at the 1989 Interservice,

Industry Training Systems Conference, a paper entitled "Ada: Experience It Again for the First Time", presented at the 1990 Tri-Ada Conference, and a paper entitled "Ada in the 90's," presented at the 1990 Interservice/Industry Training Systems Conference.

Dennis W. Meehl is the Department Manager for Special Programs Computer Systems at the Binghamton Operations of CAE-Link Corporation. Mr. Meehl started work with Link in 1977 and has held various technical positions in real-time simulation systems and software development environments. In 1986, he began investigating the use of Ada in real-time simulation and worked technical issues until 1987, when he was assigned as Department Manager. Mr. Meehl received his BS degree in Engineering at Case Western Reserve University in 1971.

ELECTROMAGNETIC PROPAGATION MODELING FOR DISTRIBUTED SIMULATION

James J. Gonzalez
BBN Systems and Technologies
Advanced Simulation
Bolt, Beranek and Newman, Inc.
Cambridge, MA 02138

ABSTRACT

As the training industry converges on standards for Distributed Interactive Simulation, the next critical need is for better representations of electromagnetic propagation phenomena. Realistic radio simulation is essential for proper representation of command, control and communication functions. Accurate radar simulation is critical for the representation of various sensor and weapon systems for many types of platforms. Electronic countermeasures and counter-countermeasures must also be incorporated to adequately reflect the growing complexity of the modern battlefield.

This paper describes some electromagnetic propagation models that have been implemented for simulations commissioned by the U.S. Army Communications-Electronics Command (CECOM) and the Defense Advanced Research Projects Agency (DARPA) and discusses the "lessons learned" from those efforts. An approach for the efficient computation of radar intervisibility and target detection is described. Finally the DIS protocol extensions that will be needed to support and extend these models are discussed.

INTRODUCTION

Electromagnetic activity on the battlefield is widespread and growing. In addition to radio communication of voice, armed forces use systems throughout the spectrum for detection, targeting and disruption of adversaries. On the simulation front, interest has expanded from basic vehicle training to evaluation of the more complex interactions, and the effect that C3 systems have on soldier performance. While simulation of basic communications has been supported in the past, a more realistic approach has become desirable. In addition, distributed simulation is now being used as a testbed for concepts in radio-based data networks, such as the CVCC (Combat Vehicle Command and Control) elements mentioned later in this work.

In this paper, two applications of electromagnetic systems modeling are considered: combat radios and radar. The discussion of combat radio simulation is based on BBN's experience with the SINCGARS radio simulation implemented for CECOM at Fort Knox, Kentucky and Fort Monmouth, New Jersey over the last two years. The discussion of radar simulation centers on our implementation of the ADATS system for DARPA under the SIMNET contract (the DARPA-sponsored program for developing and demonstrating large-scale distributed-simulation technology).

RADIO SIMULATION

Modern combat radio systems provide means not only for voice communication, but also for digital data transfer among participants. Realistic simulation of this communication path is important in training situations, where techniques for taking advantage of this resource, and for coping with its failure, can be explored in the context of an exercise.

What Went Before

In earlier SIMNET vehicle simulators, radio communication was provided by means of citizens band (CB) radios interconnected by a network of coaxial cable and controlled by silk-screened front panels resembling those of the real combat radios (RT-442). This implementation was adequate for platoon-level training, but had certain limitations:

1. It does not allow for economical voice communication between simulators at geographically distant sites. An expensive phone line must be allocated between each site for each radio channel supported.

2. The channel allocation for CB limits one to 40 channels.

3. The radios cannot readily support data transmission.

4. Communications quality and range are uniformly perfect, regardless of distance, transmission power or terrain.

5. Simulations of direction-finding receivers and RF-seeking ordnance have no way of interacting with the vehicle radios.

6. There is no way of recording the radio traffic from an exercise with proper identification of the speakers.

In the next section, we will see how these issues are addressed in the work done for CECOM.

A Better Idea

At the direction of CECOM, BBN Systems and Technologies developed a network-based simulation of the Single-Channel Ground-Air Radio System (SINCGARS) for use both in existing M1 tank simulators at Fort Knox and in a testbed configuration at Fort Monmouth. The radio simulation implemented by BBN Systems and Technologies addressed the limitations of the CB radio approach as follows:

1. The radio simulation hosts communicate over the same simulation network as the other simulators, and can therefore take advantage of the site-to-site communication (long-haul networking, or LHN) already provided.

2. Because the "radios" are digital simulations, the number of distinct channels available is virtually unlimited.

3. The carrying of data on the simulated radio channels is provided for, including interfaces to various real and simulated sources of data.

4. Playing of received voice and introduction of errors into received data are determined by a propagation model that looks at transceiver characteristics and position on the simulated terrain, as well as signals from competing transmitters.

5. Simulations of direction-finding receivers and RF-guided ordnance can monitor the protocol data units (PDU's) used by the radio simulators to communicate over the simulation network to detect emissions from the simulated radios.

6. Timestamping and identification of simulated radio broadcasts is built into the radio protocol, and is supported by the data logger (the device used at SIMNET installations for the recording and playback of exercises).

Many of the elements that comprise this realistic radio simulation can be applied to simulations of other electromagnetic systems, as we discuss later in this paper. Key among these elements is the transmission loss model. The radio simulators share the same simulation network as the vehicles with which they are associated, allowing access to position information without modification to the vehicle simulators. The position information is used by a propagation model to determine signal loss between receiver-transmitter pairs. This signal loss can then be applied to any simulated electromagnetic transmission to determine receiver capture, interference, and bit errors. Voice and data communication can therefore be subject to the same limitations as real-world systems, providing a more accurate training environment.

A Propagation Model

SINCGARS operates in the VHF band, between 30 and 88 MHz. Signals in this band are affected by terrain and atmospheric conditions, since it propagates both in ground and sky waves CECOM recommended the Longley-Rice propagation model for this application, since the operating range and terrain fall well within the constraints of the model.

Longley-Rice uses experimentally-determined data regarding the effects of atmospheric and soil conditions to specify parameters regarding general propagation characteristics. It then adds information about the terrain, in the form of a "roughness figure". The calculation involves reading the terrain elevation at fixed intervals (50 meters) between the receiver and transmitter. In practice, this value is calculated each time a simulated radio moves a significant (50-meter) distance. Finally, it deducts for spherical spreading loss (a function of distance) to produce a loss figure for a given receiver-transmitter pair.

The model is quite convincing in operation. Driving around the Fort Knox terrain in an M1 simulator, communication with a stationary vehicle comes and goes as one goes over hills, around trees and moves closer or farther away.

Radio Simulation Hardware

The radio simulation is supported by a set of simulation hosts (shown in Figure 3.1) connected to the simulation network. Each has the resources to support crew interfaces to radio controls, the network interface, and a copy of the terrain model. In the implementation done for CECOM, the host (a Concurrent model 6600) supports multiple radios attached to multiple vehicles. A later version, done under the MULTIRAD program for the U.S. Air Force's Human Resources Laboratory, supports multiple radios for a single vehicle. It runs on less expensive Motorola VME-bus computer boards, thus allowing a return to the SIMNET concept of one-vehicle-per-host.

In both the CECOM and HRL systems, voice is recorded and played using a special board called a SIMVAD (SIMNET Voice Analog-Digital). The SIMVAD contains two analog I/O channels, each with a dedicated DSP (digital signal processor) to do encoding and decoding.

Two encoding schemes are currently available. APCHQ (Adaptive Pulse-Code with Hybrid Quantization) was used on the original SIMVAD. It is well-suited to producing high-quality speech at minimal bit rates. CVSD (Continuously-Variable Slope-Delta) is less computationally-demanding, and lower-quality, but is more tolerant of the bit errors likely to be encountered in a combat situation. APCHQ is intended for use when high-quality speech is required to approximate the performance of analog radios. CVSD is used in the real SINCGARS radios, and is therefore best when simulating that radio system.

The SINCGARS simulation installed at Fort Monmouth supports connection of actual Army CHS (Common Hardware-Software) computing devices, in order to allow them to utilize the simulated radio networks for data communication. The Fort Monmouth installation can also be connected to real SINCGARS radios operating in their retransmit mode, to allow real SINCGARS radio nets to interconnect with simulated radio nets. A synchronous serial communications board supports both the CHS and SINCGARS connections.

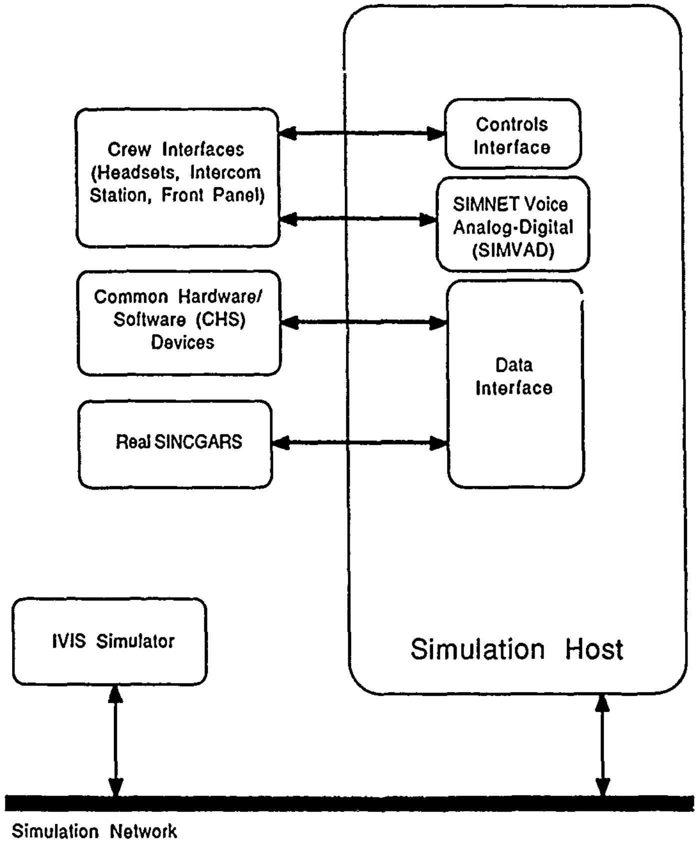


Figure 3.1. Radio Simulation Hardware Layout

A network-based interface is also provided to allow connection of IVIS (Inter-Vehicular Information System) simulators to the simulated SINCGARS radios. The IVIS simulators, which provide a graphical communication interface to tank commanders, were developed concurrently with the SINCGARS simulation for use in experiments at Fort Knox. This is an example of the testbed concept. The IVIS simulators were developed to allow the Army Research Institute to explore user interface concepts. Using a realistic SINCGARS simulation with the simulated IVIS units allows investigators to evaluate performance under the limitations it would face in the real world.

The Radio Network Monitor

The Radio Network Monitor (RadMon) is a system for monitoring the radio simulators. It is a separate computer residing on the simulation network, with a graphical display. It is capable of producing a map of the radios, and of showing their connectivity, as determined by tuning and signal loss. It records changes in the state of the radios on the network, allowing for the collection of statistics on loading of the simulated radio networks.

RadMon is useful to technicians in troubleshooting the radio simulation during exercises. Its primary application, however, is as an analysis tool for the testbed application. Because it can supply statistics regarding channel utilization, concepts in combat radio network organization and integrated voice-data protocols can be explored in the controlled environment of the simulated world.

RADAR SIMULATION

The concepts that have been applied to providing more realistic simulation of radio communication can readily be extended to other types of electromagnetic simulation, including radar systems.

The ADATS Simulation

BBN Systems and Technologies completed a simulation of the Air-Defense/Anti-Tank System (ADATS), including its radar targeting system, in 1990. It uses a propagation model specific to the system simulated in order to provide accurate performance.

The ADATS radar simulation consists of a collection of simulation hosts residing on the network with the vehicles it is targeting. The radar simulation hosts sends out Radiate PDU's, which notify other simulators that they are being illuminated and indicating whether or not ADATS successfully detects them. The PDU's are recorded by the data logger for future reference. The target vehicles, upon reception of such a PDU, directs their threat receiver to respond appropriately. The ADATS simulators themselves use four criteria to determine successful detection:

- Target Velocity.
- A roll-of-the-dice weighted by range and target vehicle type.

- A roll-of-the-dice regarding the effectiveness of jamming by the target. The simulators supporting the target vehicles do not support ECM, so this function also fell to the ADATS simulators.

- Line-of-Sight; as determined by an intervisibility function that examines the terrain.

The weighting of the roll-of-the-dice decisions is adjusted to accurately reflect the statistical behavior of the real systems.

The Intervisibility Library

For the frequencies at which the radar operates, line-of-sight is a major factor in propagation. In developing the ADATS simulation, BBN Systems and Technologies invested considerable effort in producing software to quickly and efficiently determine line-of-sight. This software is referred to as the intervisibility library.

The intervisibility library operates by drawing two rays between the "observer" (the ADATS vehicle, in this case) and the target. One ray runs from observer to the "head" of the target, while the other is run to the "foot" of the target. This is done to determine whether line-of-sight is obscured partially or fully.

The library then checks the terrain database along each ray. The terrain is organized into squares, each of which has associated minimum and maximum elevations. If the minimum elevation at a square's boundary exceeds the ray elevation at that square, then it is assured that line-of-sight is obscured. If the maximum elevation for a square is below the ray elevation within that square, absence of obstruction within that square is assured. Otherwise, the more time-consuming task of checking individual terrain components is begun. Finally, vehicles are checked to see what impact they have on line-of-sight.

In the case of the ADATS simulation, the rays are extended beyond the target to determine if they strike terrain in the background. If so, ground clutter is included in the above-described detection criteria.

The ADATS systems' issuance of Radiate PDU's opens the door for improvements. For example, target vehicles, using hardware similar to that used to support the SINCGARS radio simulation (or sharing such hardware on vehicles equipped for radio simulation), could implement protective jamming systems. With the jammers directly associated with the target vehicles, the effects of jamming could be more accurately modeled, allowing such factors as vehicle damage, operator error and more accurate terrain conditions to be taken into account. At the same time, this distributes the computational load.

Adding the propagation model to the ADATS simulation allows more accurate event-by-event simulation by incorporating the effects of terrain-dependent signal loss, interference from other transmitters (including other ADATS units) and (when available) target-based jammers and other countermeasures.

PROTOCOL IMPLICATIONS

Protocol extension in support of radio simulation involves addition of the Radio PDU, which has four variants:

- Transmitter. Used by simulation hosts to advise others of a change in transmitter state. This includes transition from idle to transmitting, increase or decrease in transmission power, and selection of new transmission frequency or hopset. The information is noted by other radio simulators to determine handling of subsequent signal variants. It is also used by the radio network monitor (RadMon) to determine radio net connectivity.

- Receiver. Used by simulation hosts to advise others of a change in receiver state. This includes transition from idle to receiving, increase or decrease in received power, or change in transmitter being received. Ignored by the other radio simulators, it is used by RadMon to determine connectivity in a given radio net.

- Signal. Carries 26-millisecond segments of encoded voice or digital data. Processed by other radio simulation hosts based on information from Transmitter variants previously received. Noted by RadMon in its radio network activity statistics.

- Intercom. Contains 26 millisecond segments of encoded voice from an M1 crew station whose intercom switch is keyed. Issued by the radio simulation host to allow logging of intercom traffic. Not a radio simulation function, but takes advantage of the radio simulator's ability to digitize all microphone input.

All four variants are recorded by the Data Logger, a system installed at most SIMNET sites to record network activity during exercises for after-action review.

CONCLUSIONS

As an enhancement to crew training, radio simulation with propagation modeling has a significant impact. Soldiers at Fort Knox responded to the more realistic communications appropriately. As a testbed for communications concepts, the potential of the SINCGARS radio simulation is just now being explored, with the installation of a facility at Fort Monmouth.

The introduction of jammer simulations, a readily-developed variation of the existing radio simulation, will allow more realistic training of radar operators, pilots and others concerned with countermeasures.

A cross-fertilization between the radio and radar approaches can improve the realism of each. The consideration of competing transmitters, and the addition of Radio PDU's will allow radar systems to better model jammers. The line-of-sight determination provided by the intervisibility library will allow radio simulation to extend into the UHF band, where Longley-Rice is not suitable

BBN's experience in the CECOM and DARPA programs provides guidance for future extensions to the DIS standard, which must represent electromagnetic radiation in a manner that can support a wide range of applications.

Realistic radio communications and radar coverage will certainly play a major role in future distributed simulations.

These applications can share the simulation network. They are designed to minimize network traffic, and to cope with network latencies and the need for a uninterrupted data flow when speech is actively being reproduced. Loading is comparable to that of vehicle simulators, and newer networking technologies, such as FDDI, are opening the way to exercises with even greater numbers of participants.

REFERENCES

[1] Rastellini, M.M., editor. Electronic Combat Principles, US Air Force Pamphlet 513, 1985

[2] Gonzalez, Jim, Pope, Arthur R., Tomlinson, Ray, Van Hook, Dan. SIMNET Simulation of Radio Communication: A Testbed for Investigating Combat Vehicle C^3I Technology. BBN Technical Report 7352. September 1990.

[3] Hufford, G.A, Longley, A.G., and Kissick, W.A. A Guide to the Use of the ITS Irregular Terrain Model in the Area Prediction Mode. NTIA Report 82-100.

[4] Bloedorn, Gary W., McDonough, James G., Crooks, Julie Rahn, Fraser, Robert E. II. FAAD LOS-F-H Functional Specification. Perceptronics Technical Report PTR-1250-11-0920-88-12A. January 1990.

[5] Military Standard (Draft), Entity Information and Entity Interaction in a Distributed Interactive Simulation,. IST-PD-90-2 (Revised), University of Central Florida, Institute for Simulation and Training. January 25, 1991.

ACKNOWLEDGEMENTS

The author wishes to thank Colin Godfrey, Linda Lovero and Richard Schaffer for their technical input and reviewing, Dick Shirley and Duncan Miller for contributing to the structure and reviewing, Ray Tomlinson for his tremendous contribution to the radio simulation development, and to CECOM for sponsoring the radio simulation study.

ABOUT THE AUTHOR

Jim Gonzalez is a staff scientist with BBN Systems and Technologies' Advanced Simulation business unit. He holds a B.S. in Computer Engineering from Boston University. Since joining the SIMNET project in April 1989, Mr. Gonzalez has worked on radio simulation for CVCC and MULTIRAD (nee ACMENET). Previously Mr. Gonzalez worked 2-1/2 years in the Sensors and Surveillance business unit, designing and fielding shipboard systems for navigation and for the acquisition and analysis of underwater acoustic data.

PACKETIZED VOICE FOR SIMULATED COMMAND, CONTROL, AND COMMUNICATION

Thomas L. Gehl
International Business Machines Corporation
Manassas, Virginia

ABSTRACT

To improve its military readiness in today's budget environment, the Department of Defense needs advanced techniques that provide effective command, control, and communication (C3) training of its personnel with fewer resources. Government, industry, and academia are working to specify the distributed interactive simulation (DIS) environment, which consists of protocol data units (PDUs) that contain information about the simulated entities, a communication architecture that provides the necessary services for networked simulation, image generation databases to represent the physical environment, and performance measures for evaluation of both the simulation and training processes. This paper discusses an innovative method of integrating voice for command, control, and communication into the DIS environment. The radio communication and digitized voice characteristics that affect the C3 training architecture will be discussed. A packetized voice architecture will be proposed that provides functional radio capabilities such as (1) selecting channels of a radio communication device, (2) receiving and listening to multiple voices on one radio channel, (3) selecting filters to emulate the radio communication signatures, and (4) providing environmental effects on voice communications. The performance issues of prototyping the packetized voice architecture and a proposed DIS PDU for packetized voice will be presented.

INTRODUCTION

Historically, the emphasis of DoD simulators has been on training the student to functionally operate tactical equipment. The simulator contained the man-machine interface of the tactical device and processed responses to the functional selections of the device. For the most part, the training scenarios were the dynamic effects of peer participants in the tactical exercise. Since the individual simulators contained the tactical scenarios, few external connections to other simulators were needed. Replication of all of the tactical possibilities derived from human interaction was too costly and complex to provide realistic collective training on a large scale. Through the use of networking technologies, simulators can be linked together to provide the force-on-force engagements in a combined arms environment. C3 functions also need to be provided to further enhance the tactical training environment.

There has been much work recently, through the Workshops for the Interoperability of Defense Simulators, on draft standards for DIS. DIS is "an exercise involving the interconnection of a number of simulation devices in which the simulated entities are able to interact within a computer generated environment. The simulation devices may be present in one location, interconnected by a Local Area Network (LAN), or may be widely distributed on a Wide Area Network (WAN)."[1] The current DIS draft standard specifies the data structures or PDUs that are communicated over a LAN and WAN to multiple simulation applications to provide state information of the simulated world. This paper will discuss how distributed interactive simulation can be embraced in providing simulated communications for C3 training.

RADIO COMMUNICATION SYSTEMS

Communication is fundamental in enabling commanders to effectively command and control their forces in any battle environment. Communication not only enables commanders to convey orders that result from the decision making process but also provides commanders with the information they need to make good decisions. To provide effective communication for battle environments, operational communication systems must be survivable, flexible, reliable, secure, and interoperable. Capabilities that operational communication systems provide to the C3 process must be understood to create an effective combined arms training environment.

A general communication system consists of information source, transmitter, communication channel, receiver, and listener (Figure 1). The information source produces a message that is either written, voice, or formatted data. The transmitter converts the message into a signal format that is suitable for the communication system. The communication channel provides the medium over which the signal is transmitted. The receiver accepts the incoming signal and converts it back to the form of the original message and presents it to the listener. The full set of characteristics of a radio communication system must be understood and simulated correctly to ensure effective and realistic training for command, control, and communication.

Typical radio transmitters frequency modulate and amplify the incoming voice messages prior to the message being fed to an antenna. The method in which a particular radio modulates, amplifies, and transmits a signal creates a signature that is unique to that particular radio transmitter. The radio broadcasts the messages to all receivers on the same communication channel. The radio receivers must demodulate the incoming signal and tune to the signal frequency that corresponds to the radio channel selected by the listener. Like the transmitter, the receiver can add noise to the signal, depending on the frequency stability of the receiver. The receiver's ability to tune onto a signal is affected not only by the fidelity of its

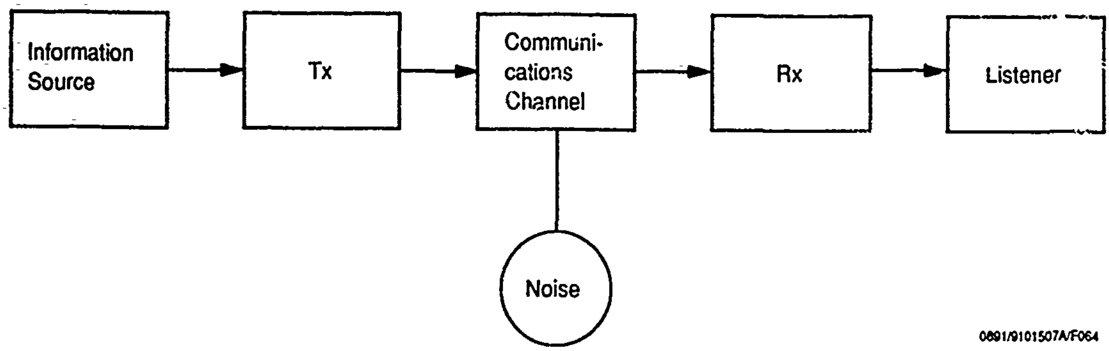


Figure 1. Communication System

tuner but also by the transmitted signal-to-noise ratio (SNR) and the propagation effects added to the signal. Receivers also provide automatic gain control that adapts to gradual changes of the signal strength from propagation attenuation, thus removing the need for continuous adjustment by the listener.

In simulating a radio communication system, we must take into consideration the communication medium's propagation effects in addition to the transmitter and receiver processing. The propagation effects of the communication medium are frequency dependent. Very High Frequency (VHF) radios, which are predominant in military applications, use a line-of-sight method of communication between the transmitter and receiver. For line-of-sight communication, the terrain and obstacles between the transmitter and receiver must be included in determining the propagation effects. Because of the frequency of VHF radios, the signal attenuation can be calculated by the spreading factor of 1/(4*PI*R**2), where R is the distance the signal travels. Most VHF radios are omni-directional, creating multi-path signals in addition to the line-of-sight signal. When a radio transmits a signal, the line-of-sight signal will combine with the multi-path signal at the receiver (Figure 2). The method in which the multi-path signal combines with the direct path signal is determined by the distances traveled by the two signals and the reflection amplitude and phase of the multi-path signal, which in turn depends on the dielectric constant of the ground.

The parameters identified in the preceding paragraphs all contribute to communications fidelity and are important in defining the architecture of the simulated communication system. Some of the parameters that are static, such as the fidelity of a particular radio, can be stored at the receive node and referenced by a radio identifier while other parameters, such as transmitter location, must be communicated with the voice information. Another consideration of the architecture is the difference between the operational and simulation communication mediums. The operational communication medium uses the atmosphere, while the simulation communication medium uses digital computer networks such as LANs.

CHARACTERISTICS OF PACKETIZED VOICE

The DIS environment is based on the interaction between simulation devices, through Local Area Networks (LANs) and Wide Area Networks (WANs), using the DIS PDUs to communicate state information of the simulated world. The DIS data is put into packets that are sent over LANs using the necessary communication protocols. One method of integrating the simulated radio communications between distributed simulation devices would be to use many of the ideas inherent to a DIS architecture. The position put forth here is to include a Voice PDU in the DIS PDU set and to utilize this new PDU to communicate packetized voice over the DIS network. In discussing how simulated communications would be provided, consideration must be given to not

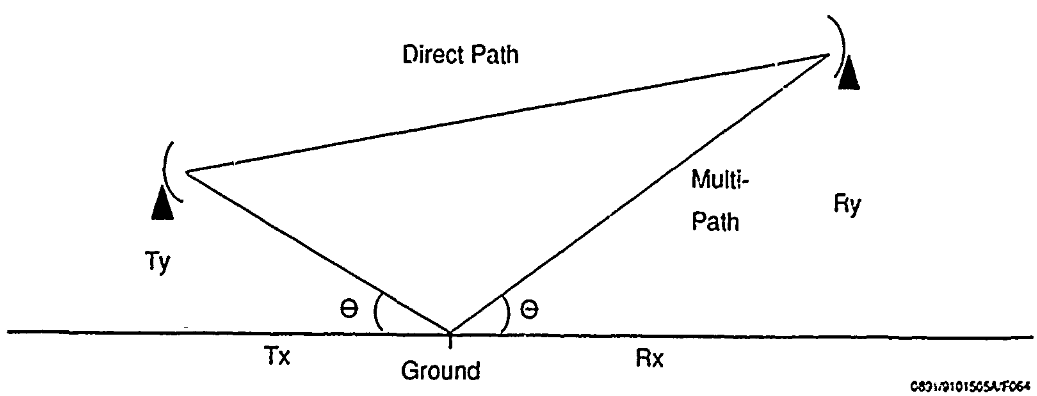


Figure 2. Multi-Path Effects of Phase and Amplitude

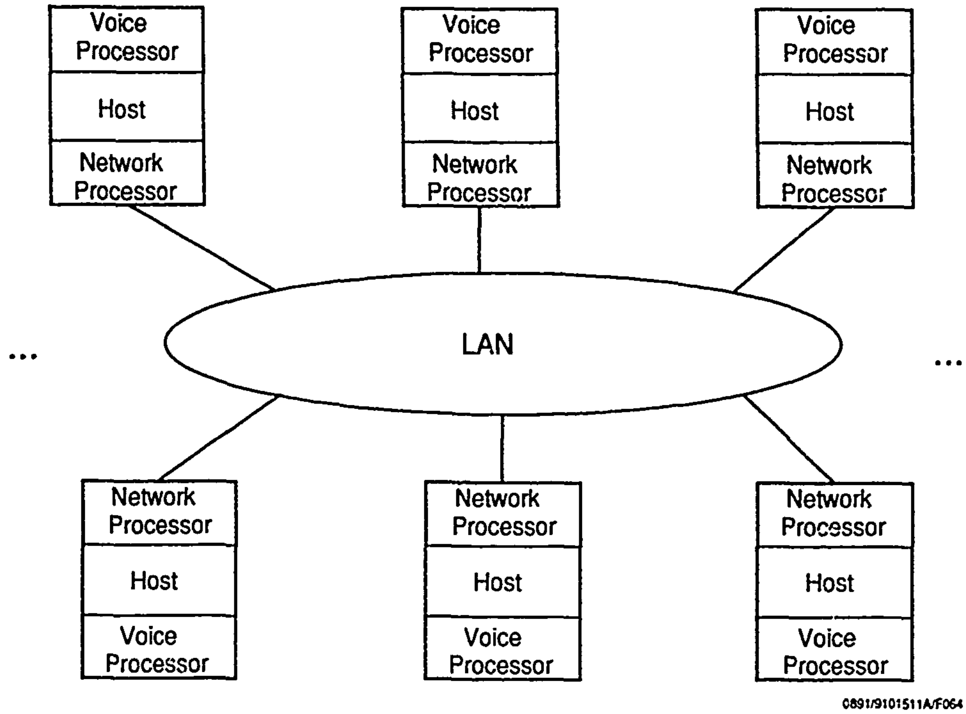


Figure 3. Packetized Voice Architecture

only the radio communication characteristics discussed earlier, but also to the packetized voice requirements.

To communicate voice information in LAN packets, we must first digitize the analog voice. Typical voice bandwidth is approximately 3.3 kHz. After adding sideguards to avoid interference between voice samples, the voice bandwidth is approximately 4 kHz. The Nyquist Theorem states that the sampling rate is required to be twice the signal bandwidth to capture all of the signal information. Thus, the required sampling rate for voice information is 8 kHz, which is the rate at which an analog to digital (A/D) converter must sample the analog voice information. The representation of the amplitude of each voice sample is determined by the number of bits used per sample. For telephone quality voice, 8 bits are used as the sample size. To provide 8 bits per sample for an 8 kHz sampling rate, the data rate for voice information must be 64 kbps. Recently, high-quality voice has been digitally communicated using 16 kbps of digital bandwidth. This reduction of digital voice bandwidth uses compressions algorithms that take advantage of redundancies in voiced vowels and consonants.

Many papers have been written recently on acceptable interspersion delays (jitter) between successive voice packets, the amount of voice information communicated per voice packet, and the end-to-end delay incurred by the voice packet. To keep the interpacket delays relatively consistent, there must be a certain level of synchronization between distributed voice nodes in communicating the voice packets. Acceptable interspersion delays between voice packets are approximately 20 milliseconds (ms). The variance in interpacket delays can be alleviated by using *play-out protocols* to smooth jitter between successive voice packets.[2] The amount of voice information per packet affects the allowable processing delay between packets and the rate at which voice packets are communicated. It is recommended that voice packets do not contain more than 50 ms of voice information per packet, and many designs involve about 20 ms of voice information per packet.[3] The overall loop delay or total delay betwen voice nodes is recommended not to be greater than 250 ms or it will affect the listener.

Digital voice does not require a very low error rate, due to the redundancy in voice information. The acceptable loss rate for voice packets through the communication medium is approximately 1-2%. This requirement is easily met by most physical communication mediums and probably is only a concern when either the receive node cannot handle the throughput requirements or when frequent collisions occur due to loading of collision avoidance protocols, such as Ethernet. Many of the above requirements not only affect the voice digitization design but also the network protocols that communicate the voice information between voice nodes.

PACKETIZED VOICE ARCHITECTURE

The voice architecture for simulated communications consists of three functional areas. (1) digitizing the analog voice to meet the minimum voice requirements, (2) processing the digitized voice at the transmit and receive nodes to provide radio communication effects, and (3) communicating the radio communication information in a method that takes into consideration both the voice and simulated communication characteristics.

The functional architecture is designed to enable integration of the simulated radio communication with the

Transmitter Signature	Transmitter SNR	Transmitter Location	Communications Channel	Compression Algorithim ID	Sample Rate	Data Sample Size	Voice Packets

0891/91001523-F054

Figure 4. Voice Packet for DIS

current DIS state information for tactical training, and consists of three functional processors: one for voice digitization, one for radio simulation, and one for network interface (Figure 3). The computational requirements and the interaction between the functional processors are affected by the complexity of providing the simulated communication effects and satisfying the voice characteristics within the DIS environment.

To digitize the analog voice, A/D converters (transmitters) are designed to have a sampling rate of 8 kHz and a sample size of 8 bits. Thus, the converters are producing 64 kbps of data to digitally represent the analog voice signal. The digitized voice is buffered in sample memory. The voice packet size for this architecture has been selected to be 2048 bits, which equates to 32 ms of voice information per packet.[4] A digital signal processor (DSP) is included on the voice adapter to control the sampling rate of the digital converters, add real world sounds to the digital voice, and process algorithms for voice compressions, if desired.

The voice processor or DSP interfaces to the host through shared memory. When a voice packet is complete, the DSP interrupts the host; subsequently, the host services the interrupt. Servicing the interrupt involves the host preparing the simulation radio packet and communicating the simulated radio packet through the network adapter. A method of preparing the simulated radio packet is to add data fields, containing state information, to the voice packets (Figure 4). This method is analogous to the DIS concept. The following information has been selected to be communicated in the data fields with the digital voice packets:

- Transmitter Signature
- Transmitter SNR
- Transmitter Location
- Communication Channel
- Compression Algorithm ID
- Sample Rate
- Data Sample Size.

In communicating the simulated radio packet, the host calls a service mechanism that is supported by the network adapter. The service mechanism, provided by the protocol on the network adapter, is dependent on both the voice and simulation needs. To simulate radio communication, the protocol needs to be able to broadcast the radio information to receivers that have the same selected channel. A subset of a network broadcast capability is the multicast capability that uses a group-addressing technique that enables the receive nodes to filter information that does not match its group address. Multicast can be used to broadcast voice packets to receive nodes with the same selected radio channel. Due to the transmission rate of voice packets, there is a need to reduce the number of packets that the receive host must process. As the number of transmit nodes increases, this problem becomes more apparent and can cause overloading of the receive node. Therefore, the group-addressing technique is important to reduce the number of receive interrupts into the host processor. As a result, multicast is the primary network requirement for DIS.[5]

To enable effective voice communications, either the network adapter or the application process must ensure relatively constant arrival rates of the successive voice packets that meets the voice requirements. Fiber Distributed Data Interface-II (FDDI-II) provides an isochronous capability that is a slotted bandwidth for 125 microsecond (us) periodic voice traffic, in addition to the synchronous capability that is supported by FDDI. An isochronous capability enables communications of voice packets every 125 us, which is the voice sampling rate. The FDDI synchronous transmission capability guarantees a maximum delay between voice packets, but does not provide a constant delay as does the isochronous capability. Isochronous communications is ideal for voice traffic while synchronous communications can be designed to ensure reliable voice communications by the network adapter. If the only communication mechanism available is asynchronous, the host can provide a form of synchronization by performing a polling loop that ensures the communication of the voice packets on a pseudo-periodic interval.

Once the network adapter receives data that is destined for its host, it interrupts the host. As mentioned, the host can use the group-addressing technique to filter messages on the network adapter that are not for the selected radio channel. The host receives the incoming data and correlates the header information relative to any additional information it has stored concerning the terrain, environment, and radio characteristics. By having the receive nodes contain static information, the simulated radio packets need to only contain dynamic information, thus reducing the network bandwidth requirements. From the data fields, the host will determine what radio effects are needed and either compute them or select filters on the voice adapter to add the effects.

After the host interprets the information, it interrupts the voice adapter and selects the effects that need to be provided to the voice signal by the DSP. The effects, which are selectable by the host, can be provided by DSP algorithms, depending on the fidelity of the radio simulation desired. Once these effects are added to the signal by the DSP, the digital voice is converted to analog by a digital-to-analog (D/A) converter. The analog signal that is heard will contain the original voice signal plus any effects due to the simulation of the radio communication.

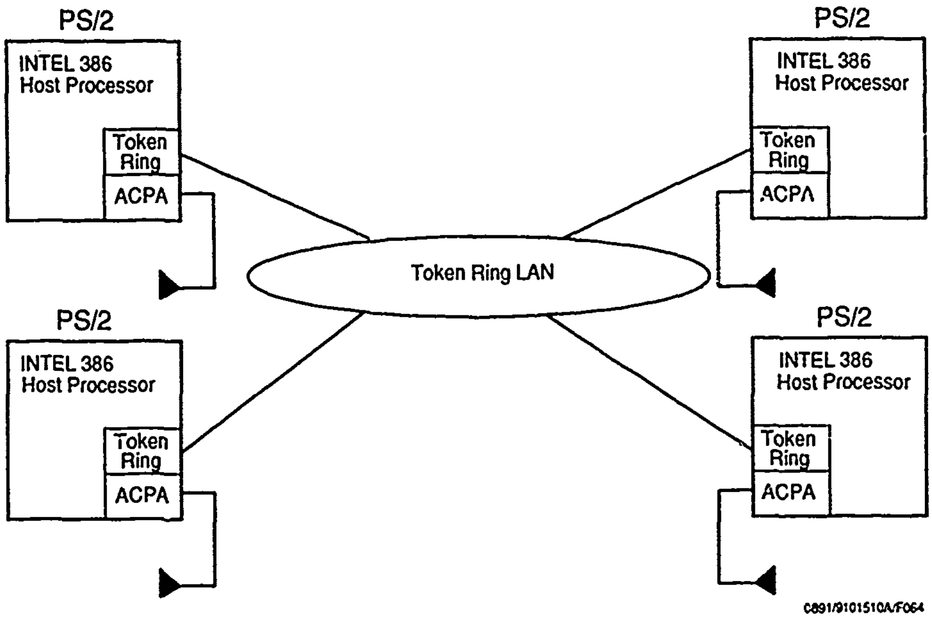


Figure 5. Packetized Voice Prototype

An example of some of the calculations required to ensure accurate radio simulations are:

- Selecting the strongest signal that results from the combination of the line-of-sight and multi-path signals
- Selecting and processing one or more DSP algorithms to replicate transmitter signature and environmental effects
- Adding signals that are received simultaneously, if the radio being simulated allows multiple voices to be heard.

PACKETIZED VOICE PROTOTYPE

The packetized voice prototype consisted of a Personal System/2 (PS/2) Model 70 hosting an Audio Capture and Playback Adapter (ACPA) and an IBM Token Ring adapter (Figure 5). The ACPA performed the voice digitization function and the token ring adapter communicated the radio communication information. In prototyping the packetized voice architecture, the simulated radio communication was integrated with the DIS state information.

The processing performed in the prototype occurred on a priority basis, using interrupt service routines to satisfy the voice interspersion requirements. The voice interrupts occurred periodically every 23 ms from the ACPA and were serviced as highest priority. Incoming simulated radio packets from the token ring adapter were serviced as the next highest priority. And the DIS state information was calculated and communicated in the time remaining between the voice interrupt service routines.

For rapid prototyping, the ACPA — a readily available multimedia device — was used as the voice adapter. The ACPA was designed to record and play back high-quality stereo sounds that contain 22 kHz of bandwidth, thus requiring a sample size of 16 bits for stereo quality and a sampling rate of 44 kHz for stereo bandwidth. The ACPA was overspecified for our voice application, but it provided the functional characteristics needed to determine performance requirements for a simulated communication system. Using the ACPA's DSP, a 61 tap lowpass filter was performed on the 44 kHz sampled data with a 4:1 decimation, providing 11 kHz sampled data without aliasing. Simulating radio communications effects would require about the same amount of processing that it took to perform the 61 tap lowpass filter. The resulting data rate of the prototype was 176 kbps per node, which is 2.75 times the bandwidth needed for a packetized voice implementation. In processing the voice signal, 256 samples of voice information was buffered per packet. Taking into consideration the 11 kHz sampling rate, the 256 voice samples resulted in 23 ms of voice information per packet transfer, which meets the voice requirements.

After the voice packets were created, the ACPA interrupted a PS/2 processor, which initiated an interrupt service routine. During the interrupt service routine, the host received the data from shared memory, appended channel information to the voice information, and issued the "SEND_BROADCAST_DATAGRAM" command using the Network Basic Input Output System (NetBIOS) software to send the voice data to the token ring network adapter. The send broadcast mechanism took approximately 3 ms to service in this prototype. The interrupt service routine needed to be completed within the 23 ms period of the ACPA interrupts, which was easily accomplished for the transmit function. The time remaining between processing the interrupt service routine and the 23 ms ACPA period was allocated to any additional processing required to integrate the DIS state information.

After the communicated packet was received onto the network adapter, the host was interrupted to receive the

voice information. The host serviced this interrupt by receiving the voice information and simulating any communication effects before sending the voice information to the ACPA card. For simulated effects, incoming voice signals were added so that multiple voices can be heard when talking simultaneously on the same channel. To add multiple voice channels, three buffers were designated in the host random access memory consisting of *new data*, *old data*, and *shared memory*. When a new packet was received, the host verified the channel of the packet and discarded the packet if it was not the channel currently selected by the host. If it was the correct channel, the host moved the packet into the new data buffer, added the new data buffer to the old data buffer, and moved the added data back into the old data buffer. After the shared memory buffer was received by the ACPA, which occurred every 23 ms, the data in the old data buffer was moved into the shared memory buffer. The addition of new data with old data was performed on multiple receive frames within the 23 ms period, depending on the number of simulated radios transmitting on the same channel. Any time available between the 23 ms voice period was used to process incoming DIS state information. This function was one example of how the simulated effects could be created.

In prototyping the integration of simulated radio communications and the DIS state information, lessons were learned about the preferred capabilities of the voice and network adapters. First, the primary concern in this implementation was not so much the bandwidth utilization of the physical network but the queuing problems due to processing the frequent receive interrupts. In fact, as the number of stations were increased, the receive node became overwhelmed in processing the voice interrupts that occurred. Also, depending on the computational complexity of the effects that were provided, the host must share its resources between the processing of the application and the servicing of the voice interrupts. The computational threshold of a 20 MHz 386 machine was approached by having to process the interrupt service routines of 4 concurrent voices from the LAN while simultaneously performing simple DIS maneuvering calculations. At this time, the loading of a 4 MHz token ring was at about 40%, predominantly voice traffic that concurred with the expected traffic for 4 voices broadcasted every 23 ms over a 4 MHz token ring. In past research, it has been demonstrated that the error rate versus loading of token ring architectures, like FDDI, is good for high loading situations. Considering the 100 MHz bandwidth of FDDI, the computational loads on the host in processing the simulated radio communications were a much greater concern than the bandwidth loads of available LAN technologies.

To enhance future designs of the packetized voice architecture, the following additional capabilities would be very useful. The sample rate of the A/D and D/A converters must be at least 8 kbps with 8 bits per sample, and it would be preferred if the sample rate was selectable to simulate applications that contain higher frequency content. To drastically reduce the number of voice interrupts, voice packets which do not meet the minimum amplitude threshold due to talk spurts of normal speech pattern should not be transmitted. Papers have been written to show that thresholding voice packets can reduce the number of interrupts by at least a factor of 2. Also, it would be preferable if the packet size could be varied relative to the system error performance. The longer that the voice can tolerate between voice packets, the less interrupts the host must process. In fact, it would be very useful if the voice adapter was a bus master so that it could communicate directly to the network adapter without having to interrupt the host. In this case, the host only would be interrupted when the voice adapter needed any information to provide the effects of the radio simulation. In addition, a multicast communication service is needed to eliminate receive interrupts for voice packets that do not match the selected receiver channel.

This paper recommended an architecture for integrating simulated radio communication with distributed interactive simulation. It also revealed some lessons learned in prototyping the packetized voice architecture and desired hardware and software capabilities to develop a DIS training system that supports simulated radio communications.

CONCLUSIONS

In response to today's limited resources, the defense community must determine innovative and cost-effective methods of ensuring military readiness. The DIS environment will enable tactical training that will enhance our military readiness in a cost-effective manner. Communications in the DIS environment, both voice and data, is a key to providing effective combined arms training and mission rehearsal. Because of the recent advances in commercial technologies and distributed systems, the DIS environment has become a reality that will bring training systems into the 21th century. Innovative utilization of these networking technologies, such as integrated packetized voice, will add training capabilities that were not feasible in the past. Many more training innovations will be possible as the technologies concerning networking and distributed systems become commercially available.

REFERENCES

[1] Military Standard (Draft), "Entity Information and Entity Interaction in a Distributed Interactive Simulation,"Contract Number N61339-89-C-0043, PM TRADE, DARPA, January 25, 1991.

[2] Cohen, D., "Issues in Transnet Packetized Voice Communication," presented at 5th Symposium Data Communication, 1977.

[3] Lines, B.M., and Hughes, D., "Performance Criteria for the Transmission of Real Time Interactive Speech Over Local Area Networks Employing the Manufacturing Automated Protocol," *Polytechnic Southwest*, Plymouth, UK.

[4] Friedman, E., Student Member IEEE, and Ziegler, C., Member IEEE, "Packet Voice Communications Over PC-Based Local Area Networks," *IEEE Journal on Selected Areas in Communications*, Vol. #7, No. 2, February 1989.

[5] Gehl, T.L., Brann, J.J., "Network Requirements for Distributed Tactical Training," 12th Interservice/Industry Training Systems Conference, Orlando, FL.

ACKNOWLEDGMENTS

The author wishes to thank Joe Brann and Jon Wist for their support in reviewing the paper and his managers, Dave Dreyer and Linda Marshall, for their support. Also, I would like to thank Kathy Kelliher for her help in getting the paper out the door.

ABOUT THE AUTHOR

Thomas Gehl is a systems engineer in the IBM training systems organization. He is responsible for an Independent Research and Development project involving real-time networking of distributed simulation. He actively participates on the Communication Architecture subgroup of the Standards for Interoperability of Defense Simulations. He has supported Naval Ocean Systems Center (NOSC) efforts in determining enhancements for the Battle Force Inport Training system. He has established a Cooperative Research and Development Agreement (CRDA) with the Aircrew Training Research Division's Armstrong Laboratory at Williams Air Force Base, which involves network research for multiship training.

Mr. Gehl holds a master's degree in Systems Engineering and a bachelor's degree in Electrical Engineering from Virginia Polytechnic Institute and State University. He presented a paper entitled "Network Requirements for Distributed Tactical Training" at the 12th Interservice/Industry Training Systems Conference (I/ITSC). Mr. Gehl was formerly responsible for the design, integration, and test of digital signal processing and beamforming algorithms for the AN/BSY-1 submarine sonar system.

Voice and Data Integration in Real-Time Simulation Networks Using a Modified FDDI Protocol

M.Georgiopoulos*, N.Christou*, M.A.Bassiouni**, M.Chiou**, J.Thompson***

Department of Electrical Engineering
University of Central Florida, Orlando, FL 32816*
Department of Computer Science
University of Central Florida, Orlando, FL 32816**
Institute for Simulation and Training
12424 Research Parkway, Orlando, FL 32826***

ABSTRACT

We investigate the issue of simulation networking using a modified FDDI protocol. The simulation devices generate data (state information) and voice (FM radio). Special emphasis is focused on the reconstruction of speech of acceptable quality from voice packets. Statistical results are collected that provide us with the average data delay, the standard deviation of the data delay and the histograms of the voice packet lengths. The average data delay is useful in determing the maximum number of data stations that the protocol can support, while the histograms of the voice packet lengths are utilized to design a successful reconstruction process at the receiver site. Statistical results are provided for the case of nonsuppressed, as well as the case of suppressed silence periods of the speech signals.

1 Introduction

Communication networks are expected to handle a variety of traffic types. *Simulation networking* refers to the networking of simulation devices, via a communication network, with the sole purpose of communicating information from one simulation device to another. Currently, simulation devices require the transmission of both *data traffic* (state information) and *voice traffic* (FM radio). It is also suggested that simulation networking should be capable of handling video traffic as well. This need will arise whenever a simulation device requests terrain data information.

Fiber-Distributed Data Interface (FDDI) is a 100-Mbps local area network standard being developed by the American National Standards Institute [1], also known as ANSI, that handles a variety of traffic types. Furthermore, FDDI has a high-bandwidth capability and the potential of supporting real-time applications, such as voice or time sensitive data. The aforementioned features suggest that FDDI will be an excellent candidate in networking simulation devices.

In FDDI terminology, traffic that is assigned guaranteed bandwidth is called *synchronous traffic*; all other traffic is called *asynchronous*. Guaranteed bandwidth is usually assigned to traffic with real-time requirements, such as voice. Although a lot of work has been conducted so far in evaluating the performance of FDDI in the presence of asynchronous and synchronous traffic (e.g., [5],[6],[7]), little or no emphasis has been focused on the special case where synchronous traffic is voice. In this paper we examine the behavior of a modified FDDI protocol in the presence of voice (synchronous) traffic and data (asynchronous) traffic. The modified FDDI protocol behaves exactly like the FDDI protocol when it comes to handling the data traffic but not when it comes to handling the voice traffic. More details about the differences of the FDDI and the modified FDDI protocols are provided in Section 3.1. The integration of voice and data traffic on a single network is an interesting problem, because these two types of traffic exhibit different characteristics that should be taken into consideration by any protocol which is responsible in handling the traffic integration.

Our primary focus in this paper is to investigate the capability of the modified FDDI protocol in transmitting voice signals between two geographically distributed stations on the network. As we mentioned above, the network will be loaded with voice and data traffic. Speech can tolerate a certain amount of distortion but it is sensitive to end-to-end delay. Although the exact amount of maximum tolerable delay is subject to debate, it is generally accepted to be in the approximate range of 100-600 ms. In order to minimize packetization of voice and storage delays it has been proposed that voice packets should be relatively short of the order of 200-700 bits, and generally contain less than 10-50 ms of speech. Finally, it has been observed that the variabilty of packet delay introduced by the protocol affects the quality of the reconstructed speech at the receiver end.

2 Overview of the FDDI protocol

FDDI is a *timed token* access protocol. The network is a ring network and stations are attached on to the network at different locations around the ring. The stations choose, in a distributed fashion, a desired *token rotation time* (TRT). Basically TRT is chosen small enough to satisfy the real-time requirements of every synchronous station (i.e., a station which generates synchronous traffic). The right to use the network bandwidth for transmission of synchronous traffic is allocated among the stations in a manner which guarantees that network capacity is not exceeded. The token is then

forced by the protocol to circulate with sufficient speed so that all stations receive their allocated fractions of capacity for synchronous traffic. This is accomplished by conditioning the right to transmit the asynchronous traffic on the fact that the token has rotated fast, so that it is "ahead of schedule" with respect to the desired token rotation time (TRT). In short, the TRT value dictates a departure schedule for the token to pass from station to station, and asynchronous traffic can be transmitted only when doing so does not cause the schedule to be broken. The protocol requires that transmission of asynchronous traffic is initiated only if the token is "ahead of schedule", but after its initiation, it is allowed to continue until completion, even if it forces the token to become late ("behind schedule"). A complete description of the FDDI protocol is provided in [1]. At this point it is worth pointing out two properties of the FDDI protocol that are mentioned in [1] and proven in [3] and [4]. These properties pertain to the real-time characteristics of the protocol.

- **Property P1:**
  The average token rotation time in the absence of failures is at most equal to TRT.
- **Property P2:**
  The maximum token rotation time in the absence of failures is at most twice the TRT.

We will revisit these two properties once more, when the reconstruction of speech signals will be discussed in Section 4.

3 Simulation Results

3.1 Model of the Modified FDDI Protocol

The modified FDDI is, as the FDDI, a fiber optic ring with bandwidth of 100 Mbps (megabits per second). For our simulations we took the length of the ring to be equal to 100 km. There are various types of overhead that need to be taken into account when the FDDI protocol is considered. These types of overhead will be taken into account for the modified FDDI protocol as well.

- *Medium Propagation Delay*: There is some time required for transmissions to propagate from one point of the ring to another. This time has been approximately calculated to be 5085 nanoseconds per kilometer distance between the two points.
- *Token Transmission Time*: This corresponds to the time required to transmit a token (24 bits) and its preamble (64 bits). It is equal to 0.00088 milliseconds.
- *Station Latency*: At each station messages pass through a buffer causing a delay of 600 nanosecond per station.
- *Capture Delay*: After a station captures the token there may be a delay before transmission actually begins. This delay represents the length of the preamble that may normally precede a packet when it is initially trans mitted. We took the capture delay equal to 64 bits.
- *Packet Overhead*: There are other types of overhead associated with the transmission of a packet, besides the preamble mentioned above (e.g., starting delimiter, destination address, source address, etc.). We took the packet overhead, including the preamble, to be equal to 160 bits.

The number of stations, N, on the ring can vary. For our simulations we examined the cases where $N = 100$ and $N = 500$. Each station generates asynchronous and synchronous traffic. The synchronous traffic has priority over the asynchronous traffic.

The asynchronous traffic per station is Poisson, independent of the asynchronous traffic generated by any other station, and of identical intensity. Asynchronous stations generate packets of information of length 1024 bits. In the terminilogy of simulation networking these packets are refered to as *Distributed Interactive Simulation Packet Data Units* (DIS PDUs). Different assumptions for the packet generation processes of the asynchronous stations can be made, where stations are not considered necessarily identical, and the packet lengths are allowed to vary. Our simulation results though correspond to the special case where asynchronous traffic is Poisson and the packet length is 1024 bits.

The synchronous traffic per station corresponds to a voice signal that is sampled with the rate of 8000 samples per second and its sample is quantized as an 8-bit number. Consequently, each station generates synchronous traffic peridically with rate of 64 kbps (kilobits per second). A voice signal can be either in a *talkspurt* or a *silence* period. Useful information (i.e., voice) is generated only during talkspurt periods. Experimental results have shown that in a typical voice signal we are in a talkspurt period only 40% of the time, while we are in a silence period the remaining 60% of the time. Talkspurts and silence periods have been modeled in the literature ([9]) as independent Poisson processes. The average duration of a talkspurt is 1.34 s and the average duration of a silence period is 1.67 s ([9]). In our simulations we focused on two distinct synchronous traffic scenaria. In the first scenario we do not take advantage of the silence periods within the voice signal, and as a result each synchronous station is a periodic source of data with rate 64 kbps. In the second scenario, the silence periods are suppressed, and consequently, each synchronous station is a periodic source of data with rate 64 kbps only during the talkspurt intervals. It is obvious, that the second scenario will impose a lighter synchronous traffic load on to the network. It is worth mentioning that under the modified FDDI protocol the length of the voice packets are allowed to vary depending on the time that it takes for the token to rotate around the network. This is the only difference between the modified FDDI protocol and the FDDI protocol. The application layer on top of the FDDI protocol (data link layer) does not give us the flexibility to change the voice packet length based on the time that it takes the token to rotate. Hence, in the FDDI protocol the voice packet lengths will be multiples of a *"minimum packet length"*, which in most cases is equal to 500-1000 bits. Simulation results of the FDDI performance under the scenario of voice packets lengths that can be multiples of a "minimum packet length" are not available at this time but they are a matter of an ongoing research effort. It is worth pointing out that under the modified FDDI protocol and at low network loads the voice packets experience low delays, while at high network loads we achieve higher efficiency by utilizing longer voice packet lengths. Similar voice packetization schemes like the one assumed for the modified FDDI protocol were also considered by other researchers in the field ([10], [11]).

In the sequel, we present statistical results for the modified FDDI protocol when TRT=5,10,20 or 50 ms. The stations were assumed to be uniformly distributed on the ring. At the beginning of our simulation and in the case where silence periods were suppressed we assumed that each station is equally probable to be found in a talkspurt or in a silence period. Most of our simulations were performed for at least 1 minute of actual network operating time, and in most cases for even longer time periods to guarantee the accuracy of the statistical results.

3.2 Statistical Results

The statistical results accumulated are the *average packet delay* and the *variance of the packet delay* for the asynchronous traffic, as well as the *histogram of the voice packet lengths* for the synchronous traffic. The histogram of the voice packet lengths is very helpful information in cases where the objective is the faithful reproduction of the speech signal at the receiver end.

A histogram is constructed by dividing the entire range of the variable's (in this case the voice packet length) raw data into equal and nonoverlapping (disjoint) intervals. Thus given the range $(a, b]$ let $(b_{i-1}, b_i]$ represent the ith interval. A counter k_i is then set for the ith interval with its initial value equal to zero. This counter is incremented by one each time a data value x (voice packet length) is found to satisfy the condition:

$$b_{i-1} < x \leq b_i$$

After all the data have been counted, the histogram of x is then approximated by the discrete function:

$$h(x) = \begin{cases} \frac{k_i}{k}100\% & \text{for } b_{i-1} < x \leq b_i \text{ for all } i; \\ 0, & \text{otherwise.} \end{cases}$$

where k stands for the total number of the data values of the variable under investigation that were observed. Consider now the first scenario where silence periods are not suppressed. In Figures 1 and 2 we show the average delay and the standard deviation of the delay of asynchronous data versus traffic load of asynchronous data, when $N = 100$ and for TRT=5,10,20,50 ms. In Figures 3 and 4 we show the average delay and the standard deviation of the delay of asynchronous data versus traffic load of asynchronous data when $N = 500$ and for TRT=5,10,20,50 ms. The traffic load of asynchronous data shown in Figures 1-4 corresponds to the cumulative traffic load generated by all the asynchronous stations on the network. An asynchronous traffic load of 0.1 in Figure 1-4 corresponds to asynchronous data generated with rate of 10 Mbps over all the stations or a rate of 100 kbps per station for Figures 1-2 and a rate of 20 kbps per station for Figures 3-4 (asynchronous stations were assumed to generate identical traffic in the model of Section 3.1). The synchronous traffic generated over all synchronous stations is equal to 6.4 Mbps or 32 Mbps when $N = 100$ or $N = 500$, respectively. The results depicted in Figures 1-4 were anticipated. Some common observations from Figures 1-4 are.

- The delay increases as the asynchronous traffic load increases, when N and TRT are held fixed.

- The delay is smaller for larger TRT values, when the asynchronous traffic load and N are held fixed. This behavior is more evident for higher traffic loads (i.e., closer to the asynchronous data throughput). This is due to the fact that larger TRT values give more opportunities to asynchronous traffic to initiate transmission. An immediate result from this observation is that asynchronous data throughput increases as TRT increases.

- The delay is larger for larger N values, when the asynchronous traffic load and TRT are held fixed. This is due to two reasons: First, for larger N, the synchronous data traffic is heavier, and secondly, for larger N, the overhead (e.g., total station latency) is larger.

The next set of Figures (Figures 5-8) show the histogram of the voice packet lengths for various values of N, TRT and asynchronous data load. As in Figures 1-4 an asynchronous data load of 0.1 corresponds to asynchronous data generated with a rate of 10 Mbps over all asynchronous stations. Referring to Figures 5-8, and to additional results, not included here due to lack of space, the following observations are pertinent:

- The histogram of voice packet lengths is almost identical for various TRT values (e.g., 5,10,20,50 ms) provided that N is fixed and the asynchronous data load is light (e.g., 0.1) and fixed.

- The histogram of the voice packet lengths is shifted to the right as N increases, while TRT and the asynchronous traffic load are held fixed. This is due to the fact that as N increases the synchronous traffic load increases and the token rotates slower, resulting in increased voice packet lengths.

- The shapes of the histograms are very similar for various N (e.g., 100,500) and TRT (e.g., 5,10,20,50 ms) values, provided that the asynchronous traffic load is in the range of light to medium traffic loads or in the range of heavy traffic loads. For light to medium traffic loads the histograms look like the ones in Figures 5 and 6, while for heavy traffic loads they look like the ones in Figures 7 and 8. The range of light to medium traffic loads and the range of heavy traffic loads depends on the specific N and TRT values. For example for $N = 500$ and TRT=10 ms the range of light to medium traffic loads consists of all traffic loads below 0.35 (i.e., 35 Mbps), while the heavy traffic loads are values close to the traffic load of 0.4 (i.e., 40 Mbps) (see also Figures 3 and 4).

- The shape of the histogram of voice packet lengths changes as we approach the maximum asynchronous data load that the network can support (see Figures 7 and 8). The changes in the histogram shape as the asynchronous data traffic increases are similar for various N and TRT values (see Figures 7,8). In the limit, as the asynchronous data traffic attains its maximum value, most voice packets have length equal to 64 kbps x TRT. For example, in Figure 7 where TRT=5 ms, a synchronous source (voice signal) generates 320 bits in 5ms, while in Figure 8 where TRT=10 ms, a synchronous source (voice signal) generates 640 bits in 10 ms. Hence, when the asynchronous data load achieves its maximum value, the histogram looks like an im-

pulse around the value of 320 bits for TRT=5 ms and around the value of 640 bits for TRT=10 ms.

- Figures 5-8 indicate that properties P1 and P2 of the FDDI protocol, which are also valid for the modified FDDI protocol, and provide us with upper bounds for the average and maximum token rotation times are overly pessimistic. This is true for lightly loaded (Figures 5,6), as well as heavily loaded networks (Figures 7,8).

Let us now focus on the case where silence periods are suppressed. In Figures 9 and 10 we show the average delay and the standard deviation of the delay of asynchronous data versus traffic load of the asynchronous data, when N=100 and for TRT=5,10,20,50 ms. Similarly, in Figures 11 and 12 we show the average delay and the standard deviation of the delay of asynchronous data versus traffic load of asynchronous data, when N = 500 and for TRT=5,10,20,50 ms As in Figures 1-8, in Figures 9-12 asynchronous traffic load of 0.1 corresponds to cumulative (over all stations) asynchronous data traffic of 10 Mbps. The observations that we made for the results in Figures 1-4 are also valid for the results in Figures 9-12. Comparing Figures 1-4 and 9-12 we see that the delays of asynchronous data are smaller when the silence periods of the synchronous traffic (voice) are suppressed. This is due to the fact that with suppressed silence periods synchronous traffic imposes a lighter load on to the network. The effects of suppressed silence periods on the asynchronous data delays are more evident as the number of synchronous stations increases (compare Figures 1,2 with Figures 9,10 and Figures 3,4 with Figures 11,12). The histograms of voice packet lengths when silence periods are suppressed are ommitted due to lack of space. They are similar though with the histograms depicted in Figures 5-8 for nonsuppressed silence periods. It is worth noting that the voice packet lengths observed when the voice silence periods are suppressed are smaller than when the voice silence periods are not suppressed. This is due to the fact that with suppressed voice silence periods the synchronous trafiic load on to the network becomes lighter and the token rotates faster, resulting in reduced voice packet lengths. The effects of suppressed voice silence periods become more evident as the number of synchronous voice stations increases.

4 Reconstruction of the Voice Signal

If voice packet delay variability is significant the reconstruction of continuous speech of acceptable quality from voice packets becomes problematic. We therefore propose the following *reconstruction process* at the receiver site. We assume that the receiver has full timing information in the form of *time stamps* to accurately determine each voice packet's delay through the network, denoted by D_v. Then, the receiver adds a controlled delay D_r to every voice packet that it receives so that the total entry-to-playout delay

$$D_t = D_v + D_r$$

is as uniform as possible for all the voice packets. Based on Property P2 of the FDDI protocol, which is also true for the modified FDDI protocol, we would be inclined to choose D_r for each voice packet such that $D_v + D_r$ =2 TRT. This way we guarantee that no voice packet will be lost and the reconstruction of the speech signal will be perfect. As the simulation results indicate though, in Figures 5-8, it suffices for all cases of asynchronous traffic loads to choose D_r such that $D_v + D_r$=TRT, at the expense of losing a few voice packets when asynchronous traffic load is heavy. As we mentioned in the Introduction, certain distortion of the speech signal is tolerable. The advantadge of choosing D_r such that $D_v + D_r$=TRT instead of 2 TRT is that we save a delay of TTRT for each voice packet. This is a significant delay saving, especially for larger TRT values. Another important observation from Figures 5-8 is that if we could estimate the asynchronous data load on the network we can choose an even smaller value for D_r without losing a lot of voice packets. For example, for N = 100, TRT=10 ms and asynchronous data load of 0.1 (see FIgure 5), we can choose D_r such that $D_v + D_r$ = 1.29 ms (corresponds to the time required to generate a voice packet of 83 bits) instead of choosing D_r such that $D_v + D_r$=10 ms (=TRT).

5 Conclusions

We have examined the performance of a modified FDDI protocol in the presence of asynchronous traffic (data) and synchronous traffic (voice). Our primary purpose was to test the feasibility of the modified FDDI protocol to integrate voice and data for the particular application of simulation networking. Our conclusion is that the modified FDDI protocol can support 500 voice stations and data stations even in the cases where some of the data stations generate heavy traffic (e.g., in Figure 3 we see that we can support 500 voice and data stations, when TRT=20 ms, even if each one of the data stations generates 90 kbps of traffic). We also observed that suppression of the voice silence periods results in significant advantages, such as smaller asynchronous data delays and higher asynchronous data throughput (e.g., in Figure 11 we see that with voice silence period suppression we can support 500 voice and data stations, when TRT=20 ms, even if each one of the data stations generates 130 kbps of traffic). We have focused our attention in providing detailed histograms of the voice packet lengths that are very valuable in determining whether some of the packetization requirements, mentioned in the Introduction, are met (e.g., 200-700 bits voice packets, less than 10-50 ms of speech in each voice packet, total voice packet delays in the range of 100-600 ms). Finally, the histograms of voice packet lengths were very helpful in designing the appropriate reconstruction process at the receiver site to reassemble the original voice signal from its voice packets (see Section 4).

References

[1] FDDI, Token Ring Media Access Control ANSI standard X3.139-1987, June 30, 1987.

[2] F. E. Ross, "FDDI-a Tutorial," *IEEE Comm. Magazine*, Vol.24, No.5, pp. 10-17, 1986.

[3] M. J. Johnson, "Proof that timing requirements of the FDDI Token Ring Protocol are Satisfied," *IEEE Trans. on Comm.*, Vol. COM-35, No. 6, pp. 62-625, June 1987.

[4] K. C. Sevcik and M. J. Johnson, "Cycle Timing Properties of the FDDI Token Ring Protocol," *IEEE Trans. on Software Engineering*, Vol. SE-13, No. 3, pp. 376-385, March 1987.

[5] R. A. Simonson, "Performance Analysis of the FDDI Token Ring," Computer Science Technical Report No. TR-88-02, University of Virginia, Charlottesville, VA, February 1988.

[6] A. Schill, M. Zieher, "Performance Analysis of the FDDI 100 Mb/s Optical Token Ring," High Speed Local Area Networks, O. Spaniol and A. Danthine Editors, Elsevier Science Publisher B.V. (North-Holland), pp. 53-74, 1987.

[7] R. Jain, "Performance Analysis of the FFDI Token Ring Networks: Effect of Parameters and Guidelines for Setting TTRT," *ACM Magazine*, pp. 264-275, 1990.

[8] T. M. Chen, J. Warland, D. G. Messerschmitt, "Dynamic Priority Protocols for Packet Voice," *IEEE Journal on Selected Areas in Comm.*, Vol. 7, No. 5, pp 632-643, June 1989.

[9] P. Brady, "A Model for Generating On-Off Patterns in Two Way Communications," *Bell Syst. Tech. Journal*, Vol. 48, pp. 2445-2472, Sept. 1969.

[10] T. A. Gonsalves and F. A. Tobagi, "Comparative Performance of Voice/Data Local Area Networks," *IEEE Journal on Selected Areas in Comm.*, Vol. 7, No. 5, pp. 657-669, June 1989.

[11] T. A. Gonsalves, "Packet-Voice Communication on an Ethernet Local Area Network," *Proceedings ACM SIGCOMM Symp. Commun. Architecture, Protocols*, Austin, TX, pp. 178-185, March 1985.

ABOUT THE AUTHORS

M. Georgiopoulos received his Ph.D degree in Electrical Engineering from the University of Connecticut in 1986. He is currently an Assistant Professor in the Electrical Engineering Department at the University of Central Florida, Orlando. His current research interests include multi-user communication networks, spread spectrum communications and neural networks. Dr. Georgiopoulos is a member of IEEE and the Technical Chamber of Greece.

N. Christou is a graduate student in the Electrical Engineering Department at the University of Central Florida, pursuing his Ph.D. His current research interests lie in the field of multi-user communication networks.

M. A. Bassiouni received his Ph.D degree in Computer Science from Pennsylvania State University in 1982. He is currently an Associate Professor in the Computer Science Department at the University of Central Florida, Orlando. His current research interests include computer networks, distributed systems, data bases and their performance evaluations. He has authored several papers and has been actively involved in research on local area networks, concurrency control, data encoding, I/O measurements and modeling, schemes of file allocation and user interfaces of relational database systems. Dr. Bassiouni is a member of the Computer Society, the Association of Computing Machinery, and the American Society for Information Science.

M. Chiu is a graduate student in the Computer Science Department at the University of Central Florida, pursuing his Ph.D. His current research interests lie in the field of computer communication networks.

J. Thompson received his BS degree in Electrical Engineering from the University of Central Florida in 1978. He is currently a Research Associate for the Institute for Simulation and Training (IST) at the University of Central Florida, Orlando. Mr. Thompson has technical responsibility for all IST activities involving computer and simulation networking.

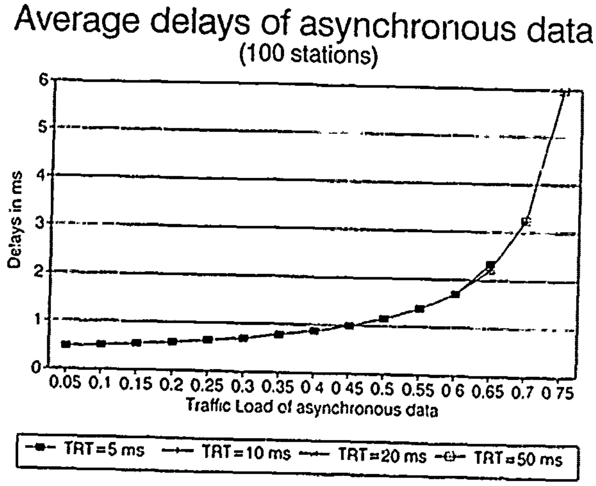


Figure 1: Average delay of asynchronous data versus traffic load of asynchronous data for N=100 and TRT=5,10,20,50 ms. Asynchronous data load of 0.1 corresponds to cumulative (over all stations) asynchronous data traffic of 10 Mbits per second.

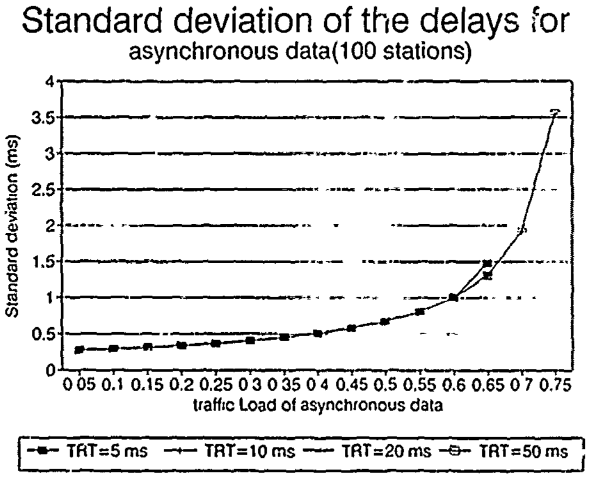


Figure 2: Standard deviation of the delay of asynchronous data versus traffic load of asynchronous data for N=100 and TRT=5,10,20,50 ms. Asynchronous data load of 0.1 corresponds to cumulative (over all stations) asynchronous data traffic of 10 Mbits per second.

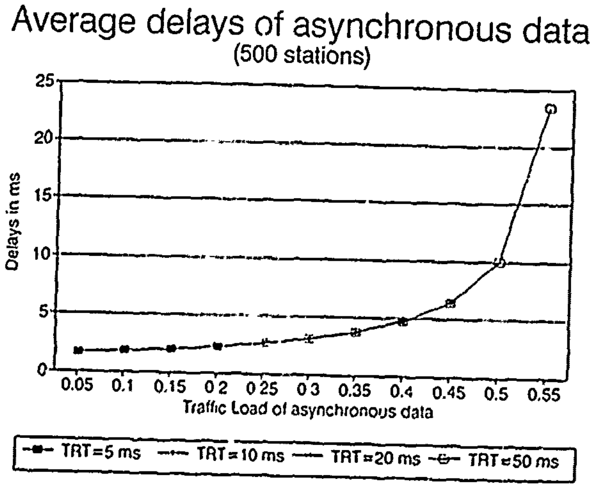


Figure 3: Average delay of asynchronous data versus traffic load of asynchronous data for N=500 and TRT=5,10,20,50 ms. Asynchronous data load of 0.1 corresponds to cumulative (over all stations) asynchronous data traffic of 10 Mbits per second.

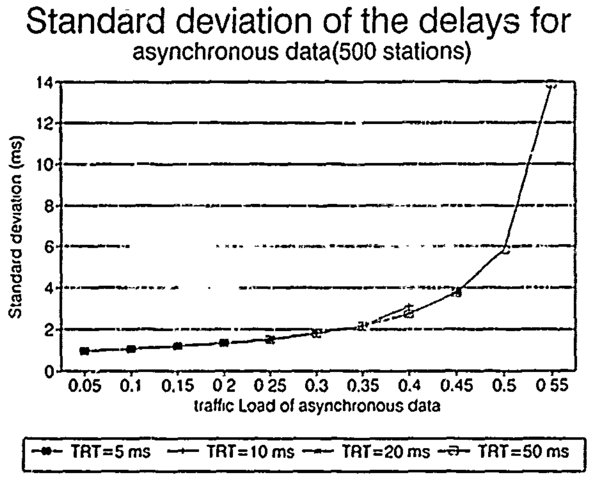


Figure 4: Standard deviation of the delay of asynchronous data versus traffic load of asynchronous data for N=500 and TRT=5,10,20,50 ms. Asynchronous data load of 0.1 corresponds to cumulative (over all stations) asynchronous data traffic of 10 Mbits per second.

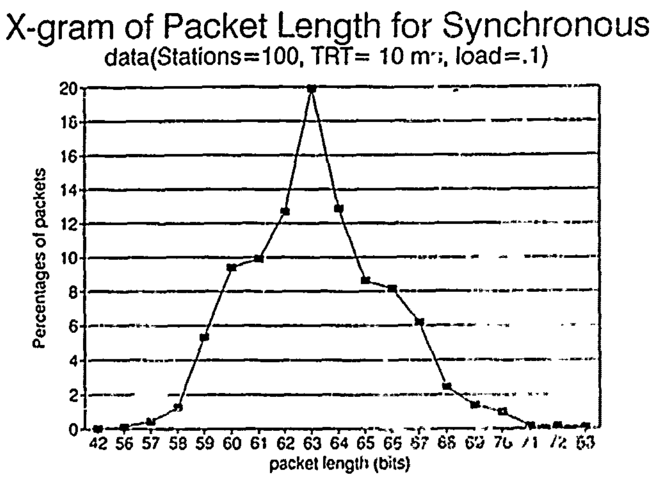


Figure 5: Histogram of the voice packet lengths for N=100, TRT=10 ms and asynchronous traffic load of 0.1. Asynchronous traffic load of 0.1 corresponds to cumulative (over all stations) asynchronous data traffic of 10 Mbits per second.

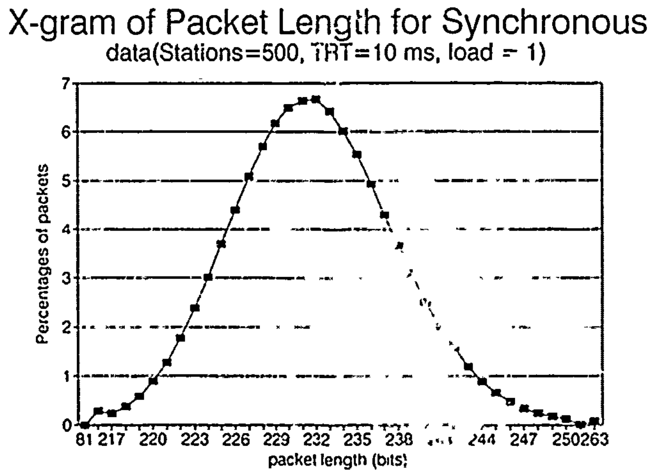


Figure 6: Histogram of the voice packet lengths for N =500, TRT=10 ms and asynchronous traffic load of 0.1. Asynchronous traffic load of 0.1 corresponds to cumulative (over all stations) asynchronous data traffic of 10 Mbits per second.

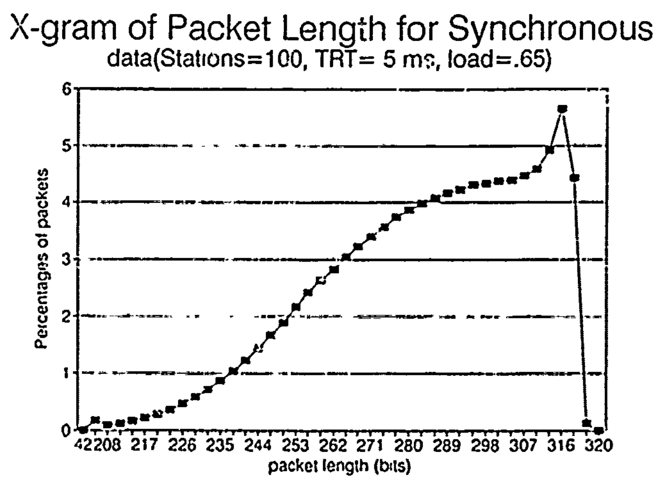


Figure 7: Histogram of the voice packet lengths for N=100, TRT=5 ms and asynchronous traffic load of 0.65. Asynchronous traffic load of 0.65 corresponds to cumulative (over all stations) asynchronous data traffic of 65 Mbits per second.

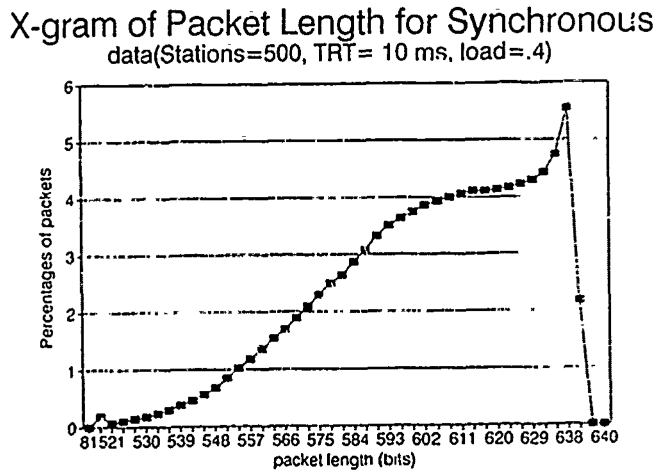


Figure 8: Histogram of the voice packet lengths for N=500, TRT=10 ms and asynchronous traffic load of 0.4. Asynchronous traffic load of 0.4 corresponds to cumulative (over all stations) asynchronous data traffic of 40 Mbits per second.

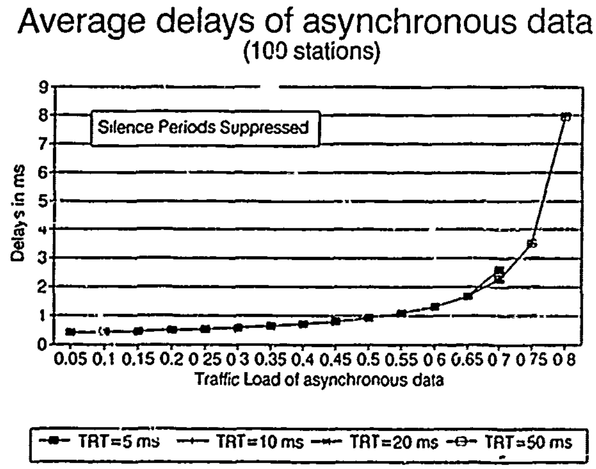


Figure 9: Average delay of asynchronous data versus traffic load of asynchronous data for N=100 and TRT=5,10,20,50 ms (silence periods suppressed). Asynchronous data load of 0.1 corresponds to cumulative (over all stations) asynchronous data traffic of 10 Mbits per second.

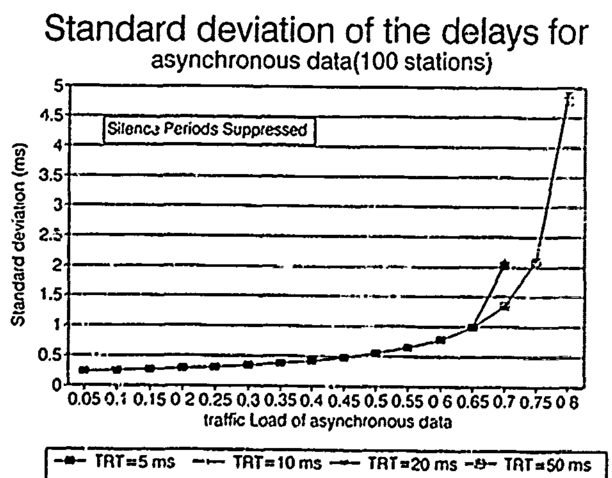


Figure 10: Standard deviation of the delay of asynchronous data versus traffic load of asynchronous data for N=100 and TRT=5,10,20,50 ms (silence periods suppressed). Asynchronous data load of 0.1 corresponds to cumulative (over all stations) asynchronous data traffic of 10 Mbits per second.

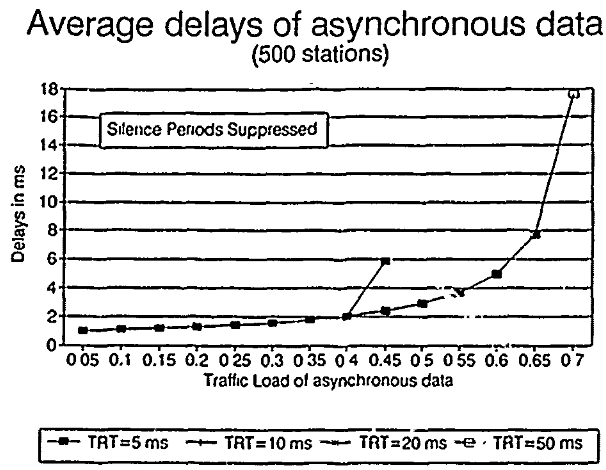


Figure 11: Average delay of asynchronous data versus traffic load of asynchronous data for N=500 and TRT=5,10,20,50 ms (silence periods suppressed). Asynchronous data load of 0.1 corresponds to cumulative (over all stations) asynchronous data traffic of 10 Mbits per second.

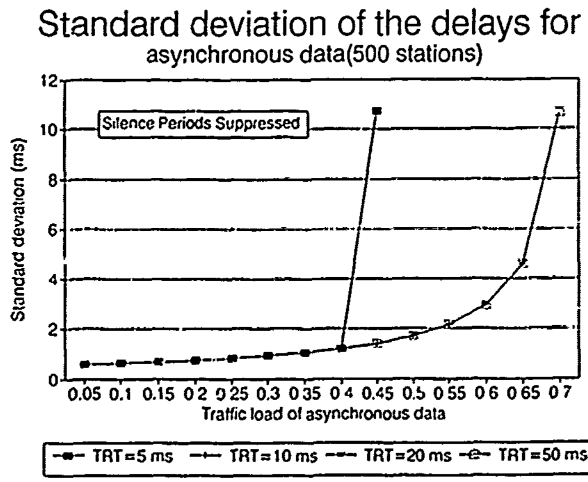


Figure 12: Standard deviation of the delay of asynchronous data versus traffic load of asynchronous data for N=500 and TRT=5,10,20,50 ms (silence periods suppressed). Asynchronous data load of 0.1 corresponds to cumulative (over all stations) asynchronous data traffic of 10 Mbits per second.

USING PARALLEL ADA IN THE IMPLEMENTATION OF SIMULATION AND TRAINING SYSTEMS

By Gary Croucher and Don Law
Encore Computer Corporation
Fort Lauderdale, Florida 33313

ABSTRACT

As simulation and training systems become more complex, vendors must rely on the ability of the target system to meet the processing needs of the application. The ever increasing complexity of today's training systems has exceeded the processing capabilities of many single CPU systems. As an alternative, more and more vendors are now considering multi-processor systems.

The Ada language is the logical choice as a software environment for developing these large scale applications. The Ada tasking mechanism can be extended to schedule and distribute tasks over multiple processors. This resulting parallel Ada runtime is capable of executing Ada tasks in parallel, while upholding the rules of the Ada language.

The decision to migrate to a parallel Ada environment is an important one involving many important factors. The intention of this study is to provide the applications developer with an insight into the specific features available in parallel Ada environments, and which features will be most useful throughout the life cycle of his application. With this information, the decision maker should be able to determine if a parallel Ada target environment is worth considering, and which types of parallel environments provide the individual features most essential to the success of his application.

INTRODUCTION

There are a number of parallel Ada environments currently available in the marketplace, each with its own set of functionality and restrictions. Unfortunately, the term "parallel" has been left open for interpretation, resulting in a conglomerate of vastly different Ada development systems being labeled as parallel. For example, it is uncertain whether an Ada development environment is to be labeled parallel because it compiles an Ada program in parallel, executes an Ada program in parallel, or compiles and executes the Ada program in parallel.

The relevant issue is not whether a particular Ada environment is or isn't parallel, or even which Ada environments have the most sophisticated parallel features. The major consideration for the developer of a large scale Ada application should be whether or not a particular parallel Ada implementation has the necessary features and performance to support and enhance the execution of his or her individual application throughout its life cycle. The ideal parallel environment for one developer may be totally inadequate for a second developer. Developers need to evaluate parallel Ada environments with respect to their own applications to determine which parallel implementation maintains the most useful set of features.

Parallel Ada environments may be capable of compiling and/or executing the target application in parallel. While parallel compilation can significantly increase the speed at which the application can be built, it has no effect on the execution speed of the application. Parallel execution (the ability of the Ada runtime system to distribute the Ada application across multiple processors to be executed in parallel), on the other hand, can have a significant impact on application performance. The remainder of this study focuses specifically on parallel runtime features that impact the execution of the application.

Clearly, the focal point for the developer is to first understand the features and attributes of parallel Ada runtimes in general, then to determine the advantages

and disadvantages of specific parallel features with respect to his application. With the information in this paper, the developer is more likely to make a more intelligent, cost-effective decision about the future direction of his or her application.

The remainder of this study is devoted to presenting a general set of features needed in parallel Ada environments used for the real-time execution of simulation and training systems. Specific advantages and disadvantages associated with some of these features, and how these features impact large-scale time critical applications, is also investigated.

PARALLELISM AND APPLICATION SUITABILITY

Before the developer decides to make the transition to a parallel Ada environment, he or she must examine the application to determine which types of parallel features would be most advantageous. The answers to the following questions typically provide good insight into how an application will perform in a parallel environment.

• Does the application utilize a large number of Ada tasks, or are runtime calls (from the underlying operating system) used to schedule and invoke concurrent code segments?

If a particular Ada runtime is parallel because it distributes Ada tasks in parallel to run over multiple processors, but the target application utilizes proprietary runtime calls, or for whatever reason, does not use Ada tasking, then this parallel implementation will not benefit the application at all. Further, applications that do utilize Ada tasks must be designed so that Ada task execution can take place in parallel. For example, an application that uses Ada tasks, but serializes execution with rendezvous, synchronization, or other methods, reaps no benefit from task parallelism.

• Does the application consist of multiple Ada tasks within a single Ada main program, or is it made up of multiple main programs communicating with each other?

If the functionality of the target application is divided up into multiple Ada main programs instead of Ada tasks, then a parallel Ada software environment is not necessary. In this case, the developer need only build and execute each main program encompassing the application using a non-parallel execution environment and assign each of the resulting executable programs to a different processor. The disadvantages to this approach are that the rich flow of control between constructs within a single Ada program is lost. Additionally, the developer is now forced to consciously partition the application himself. Finally, this approach is very inflexible. The degree of parallelism is limited to the number of main programs, and the addition or reduction of processing power requires major application rework.

• Does the application require strict priority scheduling and preemption?

Most simulation and training systems rely on preemption of executing tasks by more urgent operations, such as the expiration of a delay or an interrupt. For these systems to run smoothly using an Ada tasking model, it is usually necessary to use a runtime environment that supports strict priority scheduling. Additionally, some runtimes may even allow very high priority tasks to be locked exclusively to individual processors.

• Are fast real-time features (typically found on sequential, uniprocessor based runtimes) necessary for application execution?

Some Ada implementations support real-time features that exploit the underlying computer system. There are Ada runtime environments available with good support of real-time programs. A parallel environment to be used for simulation and training systems should not only support real-time features, but should integrate them with the parallel tasking model. These high speed features are essential to time critical applications and must not be overlooked. Some of these features may include task connections to external interrupts, optimized context switch times, and user configurable device drivers. If the application under consideration utilizes any of these relevant features, then it behooves the developer to consider parallel runtimes that incorporate some, or possibly even all, of these unique real-time features.

If the answers to the above questions indicate that a parallel implementation may be capable of providing increased application performance and additional functionality, then the next step is to evaluate typical features of parallel Ada runtimes to determine which features will be most beneficial.

In addition to systems that are in the coding or maintenance phase of their life cycle, new systems should also be evaluated for whatever type of parallel Ada execution environment is needed. An ideal situation would be for the developer to study the list of parallel features presented in the next section before beginning the high-level design of his application. In this way, the application could be designed to fully exploit the parallel features of whichever parallel Ada implementation the developer chooses.

PARALLEL RUNTIMES, FEATURES, AND FUNCTIONALITY

Assumptions

- In this section, the scope of the discussion is limited to the execution environment in which the unit of parallelism is the Ada task.
- The computer systems discussed are multiprocessor systems.
- The runtime execution environment conforms to the Ada Language Reference Manual, MIL-STD-1815.

System Architecture

The choice as to what features will be offered on a given parallel Ada implementation is ultimately driven by the underlying system architecture. System hardware can range from dual to massively parallel processors, with very tightly coupled, tightly coupled, or loosely coupled CPU/memory configurations.

A very tightly coupled multiprocessor system, where all of the processors have access to all of memory (see figure 1), is the most widely accepted architecture for parallel Ada implementations. With this architecture, task synchronization and task distribution can be controlled by the runtime kernel, which is accessible by all processors. Additionally, the rules for governing the Ada language, including the rules that dictate the scope of visibility of variables, can be adhered to without restriction, because the entire range of memory is visible to each processor in the configuration. If anything less than the entire range of memory is visible to any of the processors, then the Ada implementation either must run the risk of placing restrictions on the rules governing Ada or introduce additional overhead to support inter-processor communication. This additional overhead is typically not acceptable when introduced into large-scale time critical applications.

The applications developer should seriously consider the number of processors available on the underlying hardware before deciding whether or not to utilize a parallel Ada implementation on top of this hardware. If the processing needs of the application could increase over the life cycle of the application, then the developer should ensure that additional processors can be easily added to the existing hardware, or that compatible systems with augmented numbers of processors are available. Of course, the parallel Ada implementation should be capable of utilizing all of the available processors in the system.

Lightweight Thread Implementation

If a parallel Ada runtime environment is implemented on a multiprocessor bare machine, then the Ada environment is free to schedule the processors according to Ada semantics. One of the most desirable features of a bare machine implementation is its freedom from any constraint of an operating system scheduler. The disadvantages of a bare machine approach are that the machine must be dedicated to a single application and resources typically found on a general purpose operating system are not available.

The scheduling advantage of the bare machine approach may be captured in an Ada environment implemented on top of a real-time operating system if that operating system supports a lightweight thread model. The term "lightweight thread" refers to the ability to create additional threads of execution in a program without the overhead of creating a new process. A lightweight thread is an execution entity that is added to an existing program. It does not have its own set of files, its own address space, or the other items associated with a process. A lightweight thread consists only of a stack, a set of registers, and a program counter. Figure 2 illustrates Ada tasks implemented on traditional processes versus Ada tasks implemented with lightweight threads.

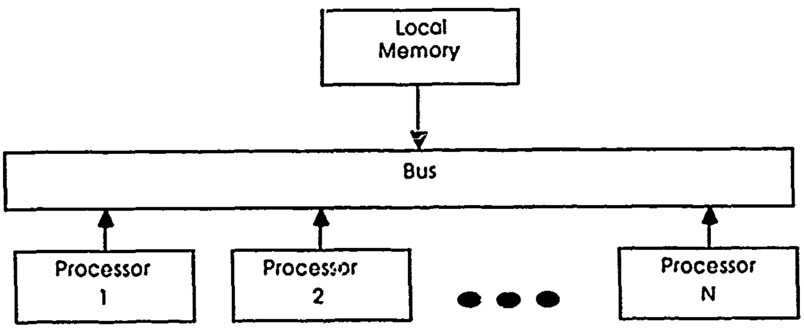


Figure 1: Multiple processors sharing one common memory

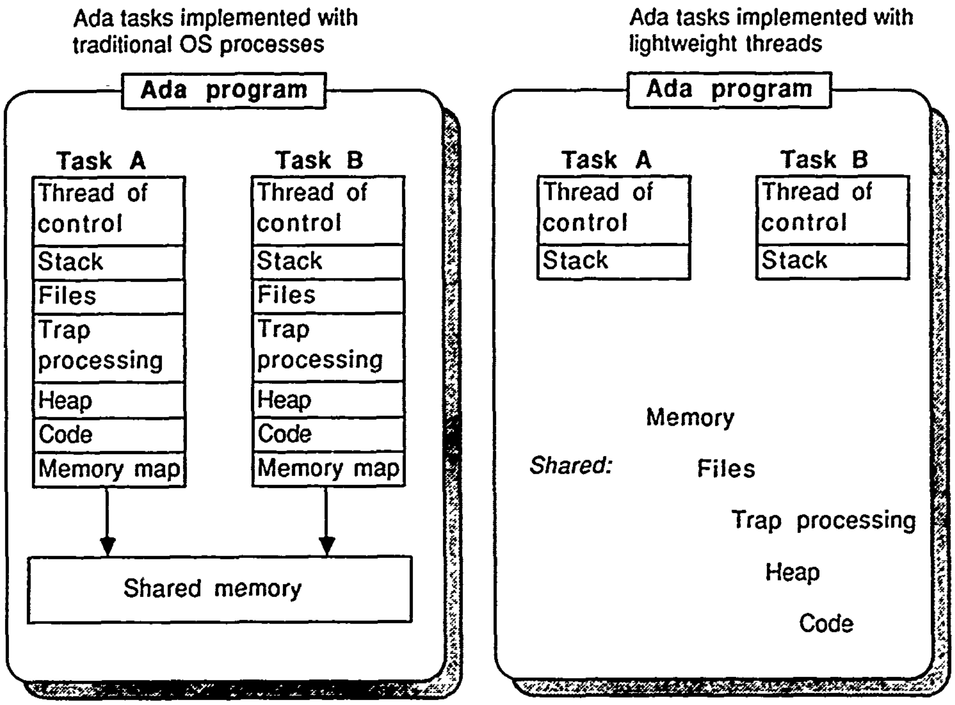


Figure 2: Traditional processes versus lightweight threads

The lightweight thread model is exactly the paradigm that is needed for Ada execution. Ada programs consist of multiple parallel threads of execution within a single program, all of which share open files, address space, trap processing, heap, and so forth.

When parallel Ada is used for a simulation system, it should be implemented on a system that matches the Ada tasking model to the computer being used. The lightweight thread model of a real-time operating system matches the needs of a multitasking Ada program.

Fine Grain versus Coarse Grain Parallelism

There are varying degrees of parallelism, even among parallel Ada runtime environments. Within the environment itself are critical regions, where tasks modify data structures shared by other tasks. The modifications must occur atomically, requiring an internal locking mechanism that is invisible to the user.

The most coarse grain environment contains a single lock for all runtime environment data structures. Whenever any task is performing any runtime action, the lock is held. In other words, the entire runtime library is considered a critical region.

Fine grain parallelism is achieved with more locking mechanisms, with locks being associated with individual data structures. This allows multiple tasks to be performing runtime system operations at the same time. Instead of contention occuring at the level of the entire runtime system, it occurs only when more than one task attempts to modify the same data structure within the runtime system at the same time.

Figure 3 illustrates the different behavior of fine grain and coarse grain environments. Time progresses from left to right in the diagram. In the coarse grain system, all four tasks compete for the same critical region, the runtime kernel. When any task is executing in the kernel, all other tasks are blocked and have to wait for the lock to be released (shown as thick arrows). This represents wasted computer resources and should be minimized. In the lower portion of the diagram, a fine grain system allows different tasks to hold locks to different resources concurrently. The only time that a task is blocked is when it needs a resource that is held by another task, illustrated by task C requesting resource X while it is in use by task B.

For an application which is not parallel or that executes on a small number of processors, a coarse grain locking system may execute slightly faster than a fine grain system because the locking mechanism has some overhead associated with it . The coarse grain system only requires a single lock and unlock operation for any runtime system call. The coarse grain system may be a satisfactory solution when a small number of processors are being used, but as the number of processors increases, the contention for the critical region increases.

Fine grain locking environments reduce the contention and therefore the blocking time of Ada

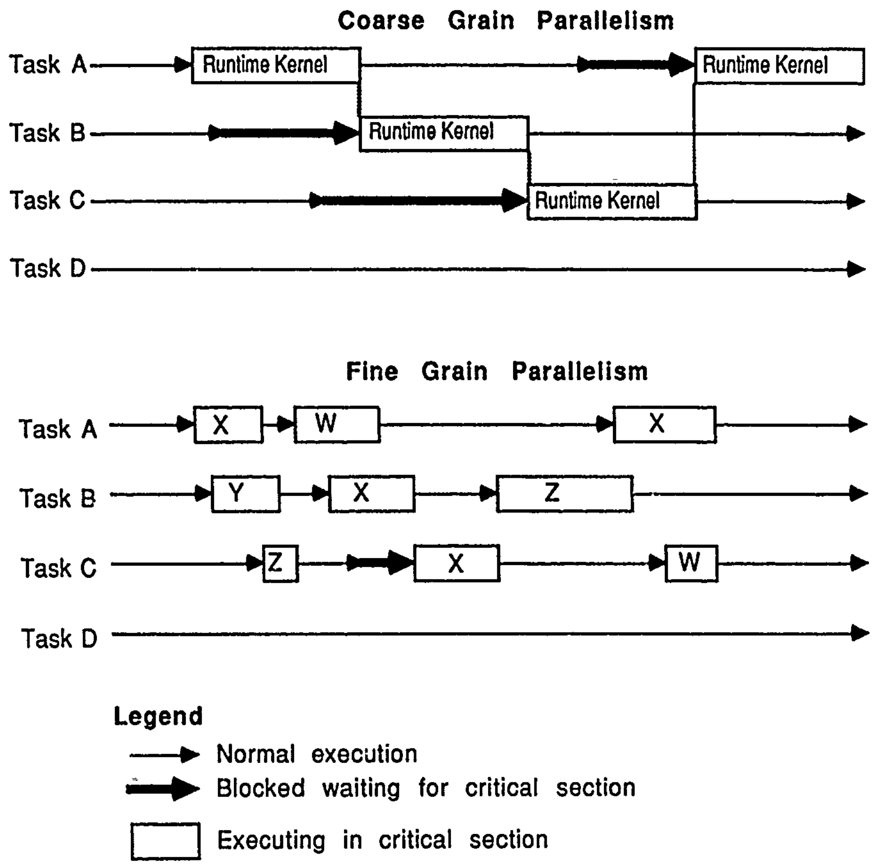


Figure 3: Coarse grain versus fine grain parallelism

tasks during runtime system operations. They are well-suited for applications that use larger numbers of processors because they use system resources more efficiently. A fine grain system may have slightly slower runtime operations than a coarse grain system because a larger number of locks must be processed for runtime operations, but the system will be more deterministic because of the reduced waiting times.

Task Priorities and Scheduling Methodology

The task priority mechanism of Ada has been a topic of much debate in the real-time systems arena. There have been several papers published on the topic that discuss some possible solutions.

A major issue is that of priority inversion on task entry queues. Ada requires that a server task execute at the priority of the client task during the rendezvous between these two tasks. This helps to reduce priority inversion, but doesn't prevent it. The priority of the server task is not affected by the client tasks which are queued waiting for a rendezvous with the server. A high priority client task may be queued waiting for a low priority operation to occur. This is priority inversion since the high priority task is held off by the execution of a lower priority task.

An example of priority inversion is shown in figure 4. The upper block shows the asks involved, with the Fire_alarm task unable to obtain service from the Comm_server task because the Check_calendar task is using the rendezvous. The example shows how the urgent Fire_alarm task is prevented from continuing execution because of the unimportant Check_calendar task. Even worse, the unrelated Sort_database task may hold off the urgent completion of the Fire_alarm rendezvous indefinitely.

The lower portion of figure 4 shows the sequence of events leading to priority inversion from left to right. When the Fire_alarm task is scheduled to run, it gains control of the processor immediately because of its priority. However is is blocked awaiting the rendezvous with Comm_server, which is executing at priority 2 (since it is serving a task with priority 2). The rendezvous may be kept from execution indefinitely by the Sort_database task, which preempts the server because of its higher priority.

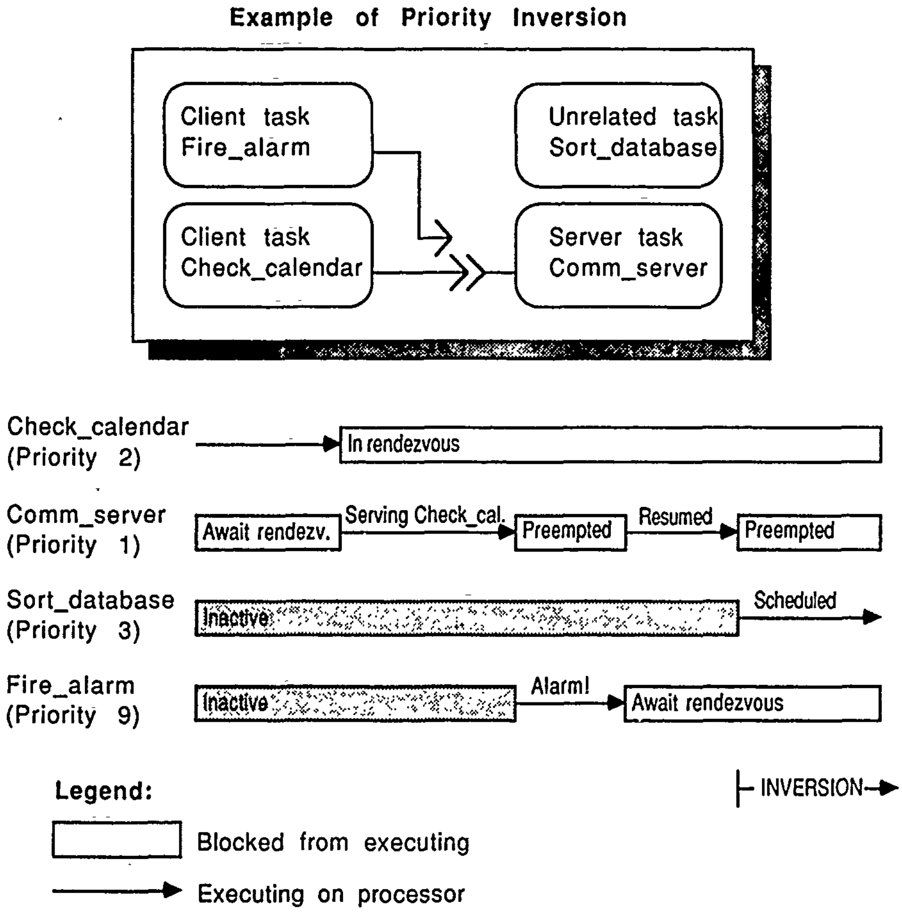


Figure 4: Priority Inversion

The problem of priority inversion is partially alleviated in a parallel Ada execution environment. With multiple parallel processors, more intermediate priority tasks must be present to starve out the low priority server. The inversion example pertains to a single processor system, but may be scaled to any number of processors by adding an unrelated medium priority task for each added processor.

The best approach when using of priorities in real-time simulation systems is to keep the model very simple. Ideally, it should be simple enough to facilitate formal proofs that priority inversion and other priority problems will not occur.

Dynamic Allocation of Processing Power

In a multiprocessor system, there is some point at which the simulation application developer specifies the number of processors that will be used to execute the target application. The longer the developer can defer this decision, the more flexible he can be with the parallelism of the application. If a non-Ada task allocation method is used, then the decision about the number of processors may have to be made as early as the design phase of the application. This is undesirable since the computer technology is likely to change before the deployment phase of the project. (see "Proprietary versus Generic Elements" above.)

The developer may be required to specify the number or processors at compile time or link time, with some directive to the compilation system or with some statements or pragmas in the source code. This requires a much lower turnaround time to change the number of processors used.

Some parallel Ada environments may allow the user to change the number of processors at the time when the program is invoked. The ultimate flexibility is the environment that allows processors to be added or removed while the program is executing. The latter feature is useful for systems that execute for long periods of time or have some fault-tolerant requirement.

Context Switch Times

Context switch times in a sequential Ada runtime are minimal, because all of the context switching and task rendezvous overhead is being carried out within a single executing process. By contrast, a parallel runtime will usually have longer context switch and task rendezvous times, because the parallel runtime

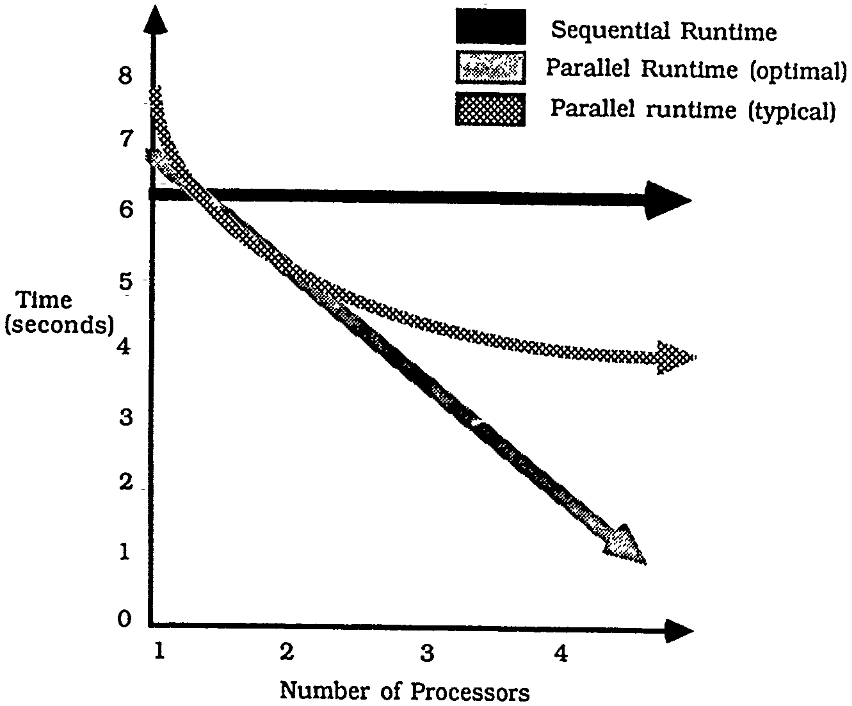


Figure 5: Execution speed of a sample application for 1 to 4 processors using a sequential and coarse grain parallel Ada runtime.

incurs the additional overhead of creating and communicating between independent threads of execution. Running on a single processor, the sequential Ada implementation will typically outperform its parallel counterpart, particularly if the application contains Ada tasks. However, as the number of available processors increase, the advantages of a parallel Ada implementation become clearer and clearer. Figure 5 is a graph of the execution of an Ada application containing a specified number of Ada tasks (in this case, more than four tasks were used). There is no interprocessor communication between tasks. The results are graphed for 1 to 4 processors when executing with a target load module linked to a sequential runtime and a parallel runtime.

The parallel runtime begins slightly slower for a single processor, due to the aforementioned overhead. However, as the number of processors increases, the parallel runtime's performance increases almost linearly. Speedup is almost linear because the application represents the optimal case (no interprocess communication and tasks immediately ready to run on available processors). A more realistic performance expectation is generated by the graph of the "typical" parallel runtime, which takes into account overhead for task initialization and communication, and assumes that there are not always tasks available to run immediately. As more and more processors are added to a parallel system, the benefits of the parallel runtime tend to diminish, until the number of processors exceed the number of available tasks and the line graph becomes horizontal.

Real-Time Features

It is difficult to implement a real-time simulation system using nothing but the generic features of Ada. However, to maximize the portability and maintainability of the system, it is best to keep the use of proprietary interfaces to a minimum.

It is good programming practice to isolate these interfaces into small areas of the system and to contain them within Ada package bodies where possible. This "information hiding" technique will make it easier to move the application to a different real-time system later in the life cycle.

The encapsulation of non-Ada real-time interfaces in packages may add some overhead to their use. The pragma "inline" may be used to reduce this overhead by causing the compiler to include the "wrapper" code directly in the calling procedure. Pragma "interface" may also be used in Ada package specifications to allow calling programs to call non-Ada interfaces.

Some simple real-time features of the system may be needed in some instances, especially where the Ada

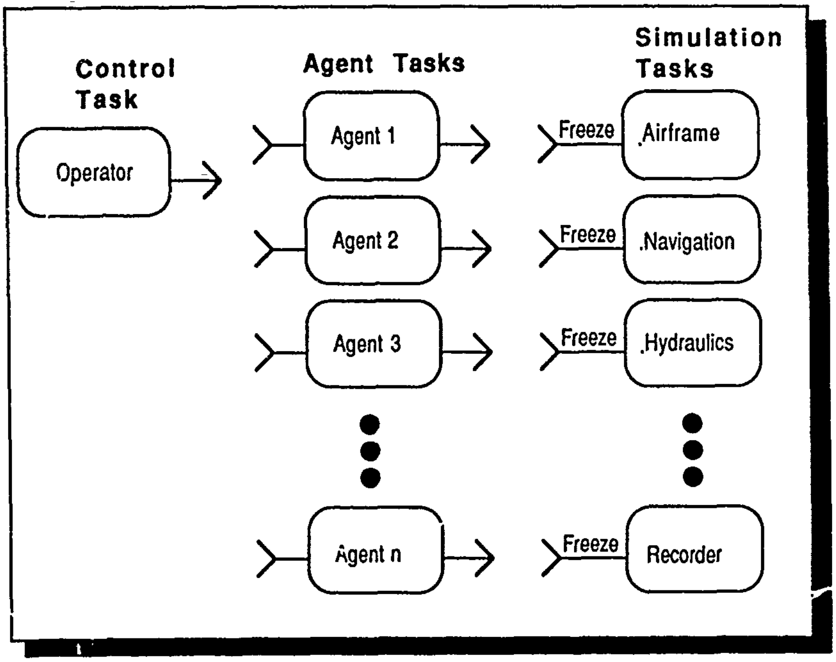


Figure 6: Solution to waitng for Freeze rendezvous

tasking model does not answer all the needs of the simulation system. One such example of this occured in a European nuclear power plant simulation and training system. The trainer required a "freeze" and "unfreeze" function, whereby the simulation could be frozen in time. Various simulation functions were needed while the freeze was in effect, so the implementation was to suspend the execution of a subset of tasks in the system. The tasks that were moving the simulation through time were to be frozen, while the maintenance tasks continued to execute.

The tasks to be frozen were synchronized by rendezvous from a master, but had "select" statements to allow the freeze operation to take place instead of the normal start of a frame. The task that performed the "freeze" operation was required to rendezvous with every task which was to be frozen. Each such rendezvous required the freeze operation task to wait until the task to be frozen reached the rendezvous before it could proceed and rendezvous with the next task to be frozen. The resulting lag caused the freeze operation to take too long to complete.

The solution to the problem was to introduce an "agent" task for every task that was to be frozen. The agent was always ready to accept the rendezvous from the freeze operation task and would then rendezvous with the task that was actually to be frozen. This solution accomplished the desired operation, but required significantly more tasks to be introduced into the system and required two rendezvouses to occur for every task to be frozen. The solution is shown in figure 6.

If a real-time supplement to suspend and resume an Ada task had been available, then the solution could have been much simpler. Using this real-time supplement, the freeze operation could have gone down the list of tasks to be frozen and issued a suspend on each one.

A basic set of real-time features should be included in a Parallel Ada system which is to be used for simulators and trainers. Besides the suspend and resume operations already mentioned, services to manage interrupts and timers, facilities for writing custom device drivers, and interfaces to real-time disk I/O are examples of other supplemental real-time services. These services could have a significant impact on the performance and success of the simulation system.

CONCLUSION

Parallel Ada development systems are an important step in the maturation of the Ada language. What was once seen only as research projects are now maturing into commercially available systems.

Some research into the high-level design of the simulation system should occur before the supporting Ada development system is selected. The implementor needs to know what questions to ask in the selection of a parallel Ada execution

implementation. There are many issues that are not apparent at the beginning of a project, but can be brought to light by looking at other similar projects that have used Ada.

The use of the Ada language does not guarantee parallelism, portability, or maintainability in itself. Such goals must be incorporated in the design of the simulation and training system. The parallelism of Ada is yet another tool, that if properly evaluated and utilized, can provide the implementor with an additional resource to help achieve his execution goals.

ABOUT THE AUTHORS

Both Don Law and Gary Croucher are senior members of the technical staff at Encore Computer Corporation in Fort Lauderdale, Florida. Both authors hold a Bachelor of Science in Computer Science and Mathematics from Furman University.

Mr. Croucher is the manager of Series 90 Ada development at Encore. He holds a Master of Science degree in Computer and Information Sciences from the University of Florida, College of Engineering. He as worked at Encore for seven years and with Apple Computers for two years prior to that.

Mr. Law has been with Encore since graduating from college seven years ago. He has worked on a prototype of the Common APSE Interface Set (CAIS) and the Ada Real-Time Executive project for Concept computers. He is currently the project leader of the Micro-ARTE project for Series 90 computers and working on a Master of Science degree from Florida Atlantic University and Carnegie-Mellon University.

Efficiency As A Part of Sound Software Engineering: Does Ada Need C?

By Marc L. Howell and Lynn D. Stuckey, Jr.

Boeing Defense and Space Group
Missiles and Space Division
Huntsville, Alabama

ABSTRACT

As high level computer languages (e.g. FORTRAN) became the required standard for new software implementation, simulation contractors began to seek exceptions for certain high utilization procedures. The contractors protested that they simply could not meet the customer's execution efficiency requirements if the language requirement was rigidly enforced. Customers frequently agreed to a marriage of convenience mixing FORTRAN and assembly language. The resulting problems of language mix helped lead the Department of Defense to develop a next generation language as the basis for all embedded computer systems, namely Ada. Since the efficiency requirements for embedded systems are even more stringent than real time simulation, one might have expected that Ada would fulfill the real time simulation speed requirements. However, as Ada has become the required simulation language in recent years, new contractor complaints about execution speed and memory usage have arisen. Contractors have sought waivers for these systems to implement certain procedures in the C language (the next generation assembly language) to improve efficiency.

The accepted truism has been that since a low level language executes so much faster and requires less memory than high level languages, then the loss to the customer of the desired features of the high order language is worth the gain in efficiency. Does this idea equally apply to applications using Ada? Is an Ada-C marriage convenient, much less in the customer's best interest? This paper presents a contrasting experience in two software applications that have traditionally been targets for language waivers: low level interface drivers and multi-dimensional interpolations. The paper discusses the specific benefits and costs of developing such applications in Ada and in C.

INTRODUCTION

Efficiency is defined as "the fact or quality of being efficient; competency in performance; the ratio of work done or energy developed by a machine or engine, etc., to the energy supplied to it". In the computer simulation world, efficiency breaks down into two basic concepts; execution speed and memory usage. To be efficient, a program must be "fast and small". In early simulation and training systems, efficiency was everything. Computers were inherently slow and possessed little memory. The program's run-time had to compensate for these drawbacks. The solution was usually assembly language, one step above the 1's and 0's. As computers progressed so did the languages, but assembly language continued to be the old standby for efficiency. So it remained in simulation until the introduction of C. Fast and compact, C provided efficiency and some of the "creature comforts" of advanced programming languages.

A few years later Ada was introduced as the new language of choice for the Department of Defense. It was primarily targeted at replacing the more than 250 languages used in military systems and to standardize the software inventory. Ada was developed with the whole software life cycle in mind. It was to address requirements analysis, production, maintenance, and reusability. Ada was not designed to be a next generation assembly language. Ada's designers assumed that processor and compiler technology would progress to the point that execution efficiency would no longer be a major concern for software development. Unfortunately, this has not always been the case. Numerous contractors have sought to use C as an efficient alternative to Ada in specific application areas. High level languages are generally considered not well suited for interfacing at the machine level or for fast execution speeds. The question arises: is the problem with the language, the compiler, or the designer? More specifically, is the question of efficiency a hardware problem or a problem with software

designers still making choices based on priorities of the past. As an answer, this paper studies the comparison of Ada and C as they were employed to develop interpolation routines and a serial I/O driver. The paper will provide a background on the software models, their development, and the results of efficiency comparisons. Also general software principals were considered in the comparison. The paper ends with a look at the future of efficiency concerns for Ada and C.

BACKGROUND

Model Choice

The candidate routines chosen for the Ada versus C comparison were interpolation routines and a serial I/O driver. Interpolation routines are used to extract operating information from a table of known values. This is done by a set of algorithms that determine a value between two known values through the assumption of linearity. These routines are used extensively in simulation and are commonly referred to as table look-ups. Their use has, in the past, been the cause of many execution efficiency problems. In many cases they must be executed hundreds of times a second, therefore, they need to be fast. In older simulations this meant that the routines were written in assembly language instead of the required application language. The test for the interpolation routines included one, two, and three dimensional data types.

The second comparison was made on a low level RS-232 serial I/O driver. As with the interpolation routines, the concern for speed of execution has driven programmers to resort to assembly language. Other important factors in I/O drivers are the needs for direct memory access and interrupts. This type of machine level interface has long been considered cumbersome or impossible in higher order languages. The drivers were designed to read and write on a serial port. The data did not have to be in ASCII.

These test cases are representative of the software applications that have historically been developed with low level languages. The two test cases selected were developed for delivery on recent contracted systems. The interpolation routines were developed in Ada for use on a flight crew trainer. The contract involved the redevelopment of existing FORTRAN code, and as such the interpolation routines were designed based on requirements and data from a earlier program. The serial driver on the other hand was developed in C as part of a control system. This control system was developed with Ada as the primary application language, while C was used for the low level I/O. In both cases, the software was developed by competent software engineers whose language of choice and expertise was the language used for development. The use of these existing software models led to a fair and more meaningful test. The fact that they were previously developed as part of a delivered product means that the software is a better representation of code actually found in industry. The original implementations are also free of any contrived problems that may have arisen from the co-development of comparison models. The development of the new routines and a closer look at the modeling details are covered in the development section of the paper.

Test Environment

The tests for efficiency were performed in an environment based on the target system for the original models. All of the code was developed on a Sun 3/260 development system running SunOS 4.0.3. The target system was a Motorola MVME-133XT (68020) running a VxWorks real-time operating system. The Ada compiler used was the VERDIX VADSWorks compiler. The C compiler was the standard C compiler delivered with Sun development systems. No physical differences existed between the two test cases for each language implementation. A synopsis of the test environment is found in Table 1.

Model	Language	Development System	Compiler	Target System
Interpolation Routines	Ada	Sun 3/260 with SunOS 4.0.3	VERDIX VADSWorks	Motorola (68020) MVME-133XT 25MHz
Interpolation Routines	C	Sun 3/260 with SunOS 4.0.3	Sun C	Motorola (68020) MVME-133XT 25MHz
RS-232 Serial I/O Driver	Ada	Sun 3/260 with SunOS 4.0.3	VERDIX VADSWorks	Motorola (68020) MVME-133XT 25MHz Z8530 Serial Controller
RS-232 Serial I/O Driver	C	Sun 3/260 with SunOS 4.0.3	Sun C	Motorola (68020) MVME-133XT 25MHz Z8530 Serial Controller

Table 1. Test Environment.

DEVELOPMENT

Interpolation Routines

The Ada interpolation routines were originally developed for use on flight trainer program. This program used the routines to determine the flight data required for the flight dynamics of a simulator. The algorithms for the interpolation routines used search techniques which were based on the last value computed. The last value was bounded by upper and lower known points. These boundaries were saved and used as the starting point for the next search, thus reducing the required search time for the next value. This approach was based on the expected application usage in flight dynamics.

The routines were optimized by the compiler and the compiled code included calls to the floating point processor. In general, the Ada implementation of the data tables required the use of variable length arrays of up to three dimensions. It was desired to improve the calling structure of the code by using array slicing to reduce the size of the arrays being passed between routines. The Ada language has very limited capabilities for multi-dimensional array slices. This required the duplication of the interpolation routines within the two and three dimensional cases. This limitation also required longer parameter lists for the Ada routines. The code structure of the Ada implementation for the interpolation program is shown in Figure 1.

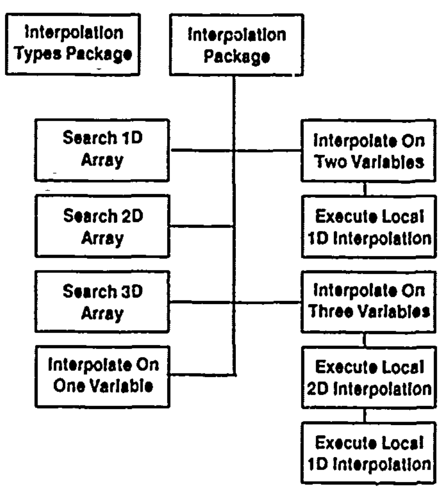


Figure 1. Interpolation Routines Software Structure, Ada Version.

The C interpolation routines were developed explicitly for this comparison. They were modeled according to the same basic requirements as the Ada routines. A number of optimizations were made in the C routines which could not be used with the Ada compiler. These included the use of register allocation and array slicing into the two and three dimensional cases. Register allocation allows the C compiler to use internal CPU registers to store frequently used values whenever possible, thus significantly reducing memory access times. As a further comparison, array slicing in the C routines was developed using two approaches, first with the actual array slices and second with pointers into the arrays. No significant impact on execution was noticed. However, it was felt that passing the actual array slices was somewhat easier to follow in the code. Since array slicing was used in the C implementation, it's structure was slightly different than the Ada. Notice the apparent lack of obvious structure with the C code structure shown in Figure 2.

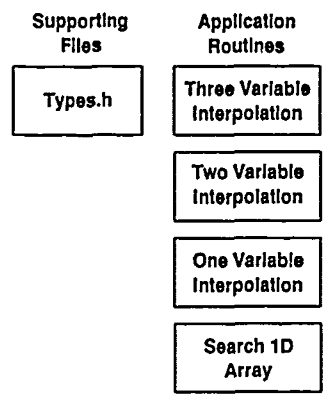


Figure 2. Interpolation Routines Software Structure, C Version.

Serial I/O Driver Routines

The C driver was written as the interface to an operator display. The required routines included a serial port initialization routine, a read port routine, a write port routine, and a routine to change the port configuration. As a standard of comparison, both implementations used the same operating system routines to achieve their purpose, i.e. VxWorks semaphores and the VxWorks I/O System interface for driver creation and calling. As with most low level I/O, it was necessary to access specific address locations in registers and memory. The C routines used declarations which defined the addresses required so that changes to the port control registers occurred

whenever the value at the address changed. The bit-wise operations built into the language allowed for easy changes to these values. The read and write routines included interrupts to notify the main procedure when the respective operation was possible. The original C code was written by a programmer experienced with writing I/O drivers in C. The original code was difficult to follow due to the extensive use of abbreviated and mnemonic variable names. The structure for the C implementation of the serial driver is shown in Figure 3. In this case the code has an implied structure, as shown by the groups of logically related functions. Unfortunately, the actual code was not grouped in this manner. Even this structure is somewhat hard to follow.

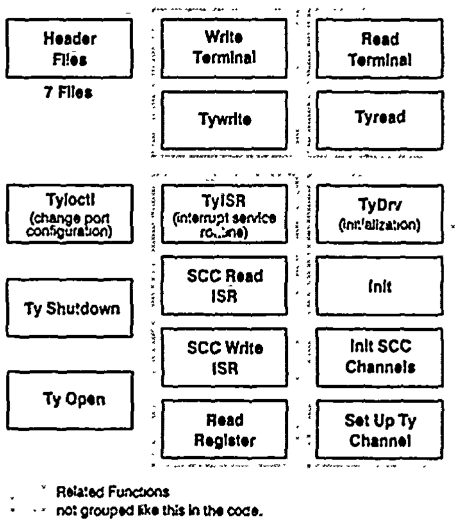


Figure 3. Serial I/O Driver Routines Software Structure, C Version.

The Ada implementation of the driver was written explicitly for this comparison. It was based on the same requirements that existed for the original C version. The program successfully used direct addressing to effect the necessary register and memory accesses. The VERDIX Ada compiler includes a set of functions which perform bit-wise operations, however, these were not used. A more desirable approach, from a portability standpoint, was to build a generic package which could be used with any integer type. This approach led to a slightly larger program size for the Ada but enhanced portability. The Ada code structure was quite different from the C. This was done as a result of the programmers experience with effective Ada packaging structure and to improve abstraction and leveling. The structure for the Ada implementation of the serial driver is shown in Figure 4.

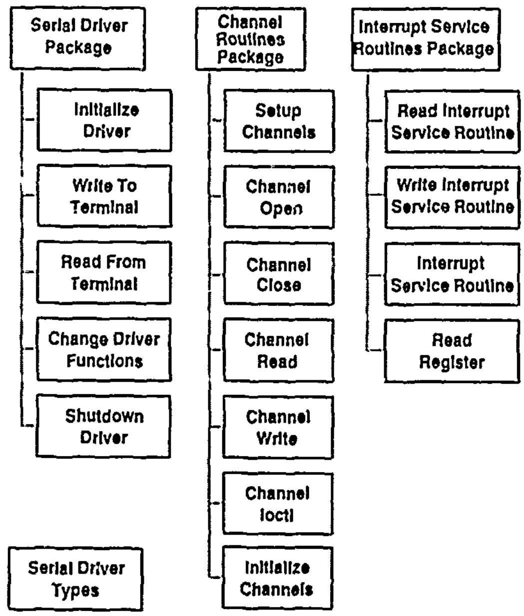


Figure 4. Serial I/O Driver Routines Software Structure, Ada Version.

RESULTS

Interpolation

The test cases for the interpolation routines consisted of several runs of different starting boundaries and desired values. The possible boundaries and the desired values included in the test are shown in Table 2. These values exercised all possible conditions for expected use of the routines. The execution times were measured using these conditions for both the Ada and C versions with the VxWorks timing function. The results were collected and ranked as worst, best, and typical (average) execution times for the one, two, and three dimensional cases. The typical case represents the most common condition for a previous value, between two known values in the table, while computing the next value, also between two known values.

Previous Boundary	Desired Value
Lower Table Limit	Lower Limit
Lower Table Limit	Interpolated
Middle of Table	Interpolated
Middle of Table	Interpolated to Next Boundary
Upper Table Limit	Upper Limit
Upper Table Limit	Interpolated

Table 2. Interpolation Test Cases.

The actual results for the one, two, and three dimensional cases are shown in Figure 5. In both language versions the best case occurred when the desired point was exactly on the upper or lower bound of the particular table searched. For this case the difference between the typical Ada and C versions is less than 5 percent. This is because the amount of code executed for this case is small and the implementations are very similar.

These internal routines acted on the entire three dimensional array with pointers to the actual desired slices. This is the primary reason for the poor performance of the three dimensional routines in comparison with C. Similar to the two dimensional case, the C version first passed a two dimensional slice and then a one dimensional slice to the respective routines. This required fewer additional parameters resulting in shorter execution times.

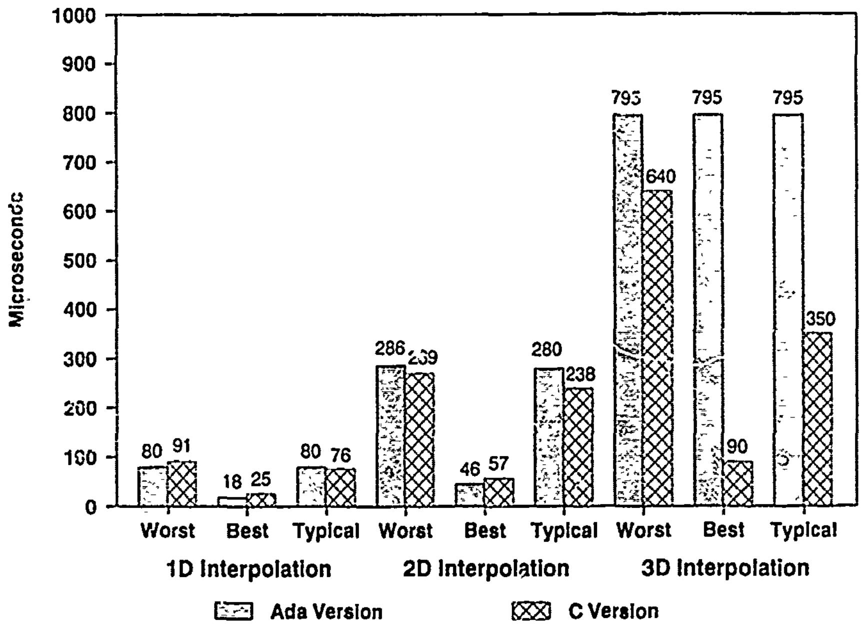


Figure 5. Interpolation Routines Speed Results.

In the two dimensional application, the variation between the two versions was greater but still fairly insignificant. For this application, the Ada version typically required 15 percent more time than the C version. This additional time in the Ada version is due to the fact that it passes the entire two dimensional array to a local one dimensional routine. It must then maintain the two dimensional indices to determine the required row from the data table. The C version uses array slicing to strip off a one dimensional slice and then passes the slice to the one dimensional routine, thus saving time and space.

For the three dimensional application, the Ada implementation required substantially more time than the C version. Again, this is a direct result of the difference between the Ada and C methods of passing array slices. The three dimensional routine used internal routines to handle the two and one dimensional interpolations it required.

The large difference in execution speed in the three dimensional case requires further discussion. Since Ada does not provide an extensive array slicing capability, the three dimensional interpolation routine could not simply call the two dimensional routine and pass it a slice from the three dimensional data array. The Ada interpolation routine used of unconstrained arrays since the size of the data array was variable. Several possible solutions to this problem were studied. The initial intuitive solution was to just declare a one dimensional unconstrained array, then declare an unconstrained of that array and so forth. The problem with this approach was that once an unconstrained array is declared it must be constrained in the declaration of the next array. Another possible solution was to declare a three dimensional array with the dimensions declared as large constants. Then an index has to be kept as to the real number of data points or the array has to be padded with zeros. This, in

fact, was the method used the method used in the C version. A brute force solution was to use representation specifications to fix the memory locations internal to the three dimensional routine. Then, through the use of the Ada Unchecked_Conversion function, create the desired slices. Based on primary design criteria of maintainability, readability, and portability these approaches were not considered effective. Thus although the solution employed is slower, it is considered a better software engineering approach in the long run.

The following information represents the relative memory sizes of the interpolation routines tested. This size consists of two numbers, the total object module size and the size of the actual executable routines excluding space for data. The compiled C routines had a total size of 1764 bytes. The actual executable code required 1466 bytes which disassembled into 367 lines of assembly language instructions. The compiled Ada routines had a total size of 75916 bytes. The actual executable code required 5538 bytes, which disassembled into 970 lines of assembly language instructions. The obvious size difference in the total object modules represents the unsuppressible overhead routines which Ada automatically provides any routine to control program errors. The difference in the actual non-error program paths (assembly instructions) may seem to be significant, however, recall that the Ada version required the duplication of routines in the two and three dimensional case. The results of the interpolation module sizes are summarized in Table 3.

	Object Module Size	Executable Code(*)	Assembly Instructions
C Version	1764 bytes	1466 bytes	367
Ada Version	75916 bytes	5538 bytes	970

(*) Excluding data space

Table 3. Interpolation Routines Module Size Results.

Serial I/O Driver

The following information represents the relative execution speeds for the serial driver routines for a total of 1000 writes. This application involved communication with a serial device, a display terminal, according to the following specification. The communication process included the channel initialization, a transmission of 29 data bytes, and the channel shutdown. The channel's transmission rate was 9600 BAUD (bits per second), utilizing 1 stop bit, 8 data bits, and no parity.

For the serial driver application, the Ada implementation required 30.93 seconds, or approximately 31 milliseconds per data transmission. The C implementation required 30.316 seconds, or approximately 30 milliseconds per data transmission. These results are shown in Figure 6. Although these execution speeds are extremely close, this could be a result of the hardware limitations of the serial port or the BAUD rate rather than differences in software. If this is the case, it is interesting to note, that the Ada does not appear to limit the hardware significantly more than the C.

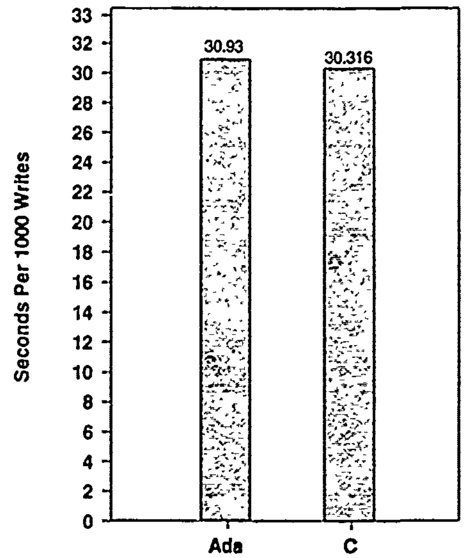


Figure 6. Serial Driver Routines Speed Results.

The following information represents the relative memory sizes of the serial driver routines. These sizes consists of the same two numbers as before, object module size and the size of the actual routines excluding space for data. The results are shown in Table 4. The compiled C routines had a total size of 4742 bytes. The actual executable code required 2174 bytes which disassembled into 560 lines of assembly language instructions. The compiled Ada routine had a total size of 62835 bytes. The actual executable code required 2380 bytes, which disassembled into 554 lines of assembly language instructions. The obvious size difference in the total object module represents the Ada overhead discussed previously. Again, notice that the differences in the actual non-error program paths is insignificant.

	Object Module Size	Executable Code(*)	Assembly Instructions
C Version	4742 bytes	2174 bytes	560
Ada Version	62835 bytes	2380 bytes	554

(*) Excluding data space

Table 4. Serial Driver Routines Module Size Results.

In general, the results from both test cases show that the use of Ada does not significantly reduce the speed efficiency of the application. The problems with array slicing in Ada are specific to the application and the design approach. The greatest drawback in the use of Ada is probably the large amount of memory required for overhead. This drawback could be a concern for applications with very limited memory.

GENERAL OBSERVATIONS

It should be noted that the routines in question were not designed with a complete emphasis on efficiency. The models were considered to be designed and coded in a manner consistent with good practices for the respective language. Software models in Ada and in C can be designed and coded similarly if not to look exactly alike. However, this would not produce representative code. The code yielded information for several other general points. Besides efficiency, the models were evaluated in terms of structure, language feature usage, maintainability, and portability. Ada is a very structured language. It is meant to encourage design and penalize poor structure. C is meant to allow quick and efficient code. C prides itself on its non-structure and its ability to "hack" a working design without regard to the quality of the layout. That is not to say that C code is not designed, but rather that it does not require or encourage sound software engineering. The main purpose of this paper was not to evaluate the software models across the entire spectrum of software engineering principles. Thus, the general observations are just that, general.

Structure and Language Usage

Several general observations were made regarding each of the languages used in this analysis. These observations had a direct effect on the methods and structures used. The most prominent ones are listed here.

[1] C does not allow multi-dimensional variable length arrays. Variable arrays are used in the Ada version of the interpolation routines to reduce the number of different sized array types needed. The C version required a different array for every case. Another possible method was to declare an array type based on some maximum size.

[2] C does not support multi-variable returns from procedures. This is due to the fact that everything in C is a function. The solution was to declare a structure which contained a variable location for the desired return values. This approach creates hidden data. The Ada version was more direct. The desired output values are listed in the procedure parameter list.

[3] The use of some C language constructs make the code less readable, for example, conditional assignment statements, extensive use of pointers '*', overloading symbols for various types, mixed type computations, and computed assignments in "if-then" statements. Such constructs are frequently used by experienced C programmers, however, usually only the person who wrote the code can easily understand what they do.

[4] Ada does not support the efficient use of array slicing. Nor does Ada support access types to multi-dimensional unconstrained arrays. C's ability to use pointers made array slicing simpler.

[5] Ada types packages were used to collect all commonly used types in the code. This feature can be very effective on large programs with many hundreds of types. C does provide a method of packaging through header files. However these are not always used efficiently.

[6] Both languages can be written to support readable and understandable code. These benefits can be enforced through coding standards. However, programmers have used mnemonic names for so long that they think mnemonics must be used for good code. This was the case with the serial driver. The C version used many mnemonics while the Ada version did not. The differences in readability were very obvious.

[7] Ada packages were also used to collect like functions and help provide leveling and abstraction. The C routines seemed to mix the functions randomly or according to calling sequence.

[8] The Ada routines could have used representation specifications and unchecked conversion to force a form of array slicing. This was considered more complex and generally 'ugly' software engineering for Ada and thus was not employeed.

Maintainability

The true cost of a simulation is only realized during the full life cycle of the software. The initial cost of development, including hardware, is minor when compared to the cost of maintaining the software. Maintainability is measured by the ease of affecting a controlled change to existing software. For these test cases, the issue of maintainability was significant for the developers of the new models. Even though the original code was less than a year old, the C serial driver had inadequate documentation and the code was almost impossible to interpret. As a result, requirements analysis for the new design of the corresponding Ada model was delayed. C code is best suited for software that is not meant to be maintained. The development effort of the Ada model was not significantly shortened by the use of the C routines. It should be made clear that the C code for the serial driver was not poorly written. It was a good working set of code that satisfied the requirements. But, the C did not encourage the design of long term maintainable code. A different story was encountered with the interpolation routines. The Ada routines provided an understandable set of requirements. The C interpolation routines, as a consequence, were simply a matter of programmer speed to develop.

Portability

Portability is the ability to transport software between computers, people, projects, and companies. Some sources refer to only the first quality as portability, and label the last three aspects 'transportability'. In the context of these tests, the edge in portability must be given to Ada. Ada is a standardized language, which in itself provides a large portion of portability that C cannot. Ada is the same from machine to machine and compiler to compiler by design. This is a problem in C, where the language might be 'normal' C, ANSI C, or even object-oriented C++. C compilers are inherently optimized for the machine on which they is hosted. This is not a bad trait for certain software products that are developed for use only on a specific machine. But in the simulation and training arena, software must be able to be adapted as the program life cycle evolves. This means that it must be able to withstand software upgrades as well as hardware upgrades. C traditionally has not directly addressed these concerns.

CONCLUSIONS

BOTH versions of the compared routine. performed their required task. There were no underlying problems with language choice that degraded the operation of the routines. The C routines were on the average more efficient. This is not that surprising since C was designed with the Motorola 68000 family of processors in mind, which was the target processor in this analysis. The interesting observation is that the C routines were not significantly more efficient, except in the case of the three dimensional interpolation routines. The Ada routines provided some side benefits along the line of maintainability, portability, and readability. This is to be expected since this was a design goal of the language and a design criteria with Boeing Ada development.

To the question of whether or not Ada needs C, the answer must be 'not really'. The C code was more efficient, but not so much as to warrant the problems of requesting a language waiver, having proficient programmers (developers and maintainers) in two languages, and the added cost of two compilers and development tools. In an environment where Ada is not a mandate, then there are surely applications that would benefit from the usage of C. But if Ada is required, as it now is with all DOD programs, then these results show that the program can be implemented in that language alone and will not require a "more efficient" language (C or Assembly Language) in order to meet requirements.

The gap in efficiency is shrinking more and more as the Ada market and programming capability matures. Ada is still relatively new in terms of comparison to FORTRAN or even C. An example of new innovation has recently been developed by Tartan, Inc. Tartan has developed an Ada compiler optimized for the TI 320C30 digital signal processor. Although this is a specialized processor, Tartan was able to significantly reduce the size and execution speed

of the object code compared to the C version for this processor. The results of this effort produced Ada code that was anywhere from 22% to 336% faster and with a 20% reduction in module memory size. Similar work to improve the efficiency of Ada is being done with the Intel 80960 processor which promises to improve Ada execution rates even more. These trends will help move Ada away from the stereotype of an inefficient high level software language.

REFERENCES

(1) G. Booch, *Software Engineering with Ada*, Second Edition, Benjamin/Cummings, Menlo Park, California, 1987, p. 31.

(2) Ann S. Eustice, "Tracking the Practical at TRI-Ada", Ada Information Clearinghouse Newsletter, Vol. IX, No. 1, March 1991, p. 12-16.

(3) Brian W. Kennighan and Dennis M. Ritchie, *The C Programming Language*, Second Edition, Englewood Cliffs, New Jersey, Prentice Hall, 1988.

(4) Daniel A. Syiek and Daniel Burton, "Optimization Delivers Fast, Compact Ada Code for the TI 320C30 Digital Signal Processor", Tartan Laboratories, Inc., Monroe Pennsylvania.

(5) Tartan, Inc., "Tartan Ada Outperforms Latest C Compiler on the TI 320C30 DSP", (News Release), December 5, 1990.

(6) VERDIX Ada Development System VADS-Works, Version 5.5, for Sun-3 UNIX to Motorola 68000 family, Reference Manual, Verdix Corp., 1988.

ABOUT THE AUTHORS

Marc L. Howell is a software systems engineer with Boeing Defense and Space Group in the Missiles and Space Division. He has been responsible for software design and analysis on the Space Station program and several other Boeing projects. He is currently working on research and development studies for improving existing software implementations by using Ada software development methodologies and techniques. Mr. Howell has a Bachelor of Science degree in Electrical Engineering from the University of Alabama in Huntsville.

Lynn D. Stuckey, Jr. is a software systems engineer with Boeing Defense and Space Group in the Missiles and Space Division. He has been responsible for software design, code, test, and integration on several Boeing simulation projects including the Ada Simulator Validation Program and the Modular Simulator System. He is currently involved in research and development activities dealing with software development for weapon and threat simulators. Mr. Stuckey holds a Bachelor of Science degree in Electrical Engineering from the University of Alabama in Huntsville.

DO YOU SEE WHAT I SEE? INSTRUCTIONAL STRATEGIES FOR TACTICAL DECISION MAKING TEAMS

Janis A. Cannon-Bowers, Ph.D.
Eduardo Salas, Ph.D.
Catherine V. Baker, M.A.

Naval Training Systems Center
Orlando, FL

ABSTRACT

Military tasks often require the coordinated effort of a team of operators for successful execution. In tactical decision making situations, team members must gather, integrate and communicate crucial information in support of decisions where an incorrect response can have catastrophic consequences. Therefore, a viable goal of training for tactical decision making teams must be to improve the quality of teamwork and team coordination. It has been argued recently that the nature of teamwork and coordination behavior can be understood in terms of mental model theory. The notion of "mental models" has been invoked as an explanatory mechanism by those studying skilled performance and system control for a number of years. With respect to training, several researchers have suggested that the goal of instruction should be to foster accurate mental representations of the task. It is contended in this paper that the mental model construct may be particularly useful in developing team training strategies and understanding the nature of teamwork. Specifically, the ability of teams to coordinate activity and adapt to task demands in absence of overt communication opportunities may be hypothesized to be a result of shared mental models of the task and team among members. A rationale for adopting the shared mental model hypothesis is presented, along with the implications of such a position for training design.

INTRODUCTION

Critical performance in many complex systems depends on the coordinated activity of a team of individuals. Military teams, in particular, must operate in situations where ineffective performance can have disastrous consequences. Despite a considerable amount of research into the area of team performance and team training, however, relatively little is known about how to train teams or to manage team performance effectively. This is particularly true in the area of team decision making where teams must gather, process and integrate information in support of a decision. Recently, several authors have suggested that team performance can be understood in terms of shared mental models of the task and team among team members [5, 17, 18]. The purpose of this paper is to show how the mental model construct has the potential to advance understanding of the nature of teamwork and development of team training interventions. To accomplish this goal, the areas of mental model research and team performance research will be introduced and reviewed briefly. Following this, the notion of shared mental models will be discussed, including a description of its utility in explaining teamwork behavior and its implications for team tactical training system design.

MENTAL MODELS

The notion of "mental models" has been invoked as an explanatory mechanism for a number of years by those studying skilled performance and system control [26, 12, 22]. According to Rouse and Morris [22], a mental model can be defined as a "mechanism whereby humans generate descriptions of system purpose and form, explanations of system functioning and observed system states, and predictions of future system states" (p. 360). In the area of cognitive psychology, researchers have suggested that mental models are important to the understanding of how humans interact and cope with the world [22]. For example, Williams, Hollan & Stevens [29] maintain that mental models allow people to predict and explain system behavior, and help them to understand the relationship between system components and events. Wickens [28] contends further that mental models provide a source of people's expectations. In an even more general view, Johnson-Laird [13] suggests that people "understand the world by constructing working models of it in their mind" (p. 10). Mental models enable people to draw inferences and make predictions, to understand phenomena, to decide what actions to take, to control system execution, and to experience events vicariously [13].

Rouse and Morris [22] concluded that a number of common themes can be drawn among theories that describe the purpose of mental models; namely that mental models serve to help people describe, explain and predict system behavior. It must also be noted that most theorists conceptualize mental models as more than simple mental images. Instead. mental

models are manipulable, enabling people to predict system states via mental manipulation of model parameters (see Johnson-Laird, [22] for a detailed description of mental model functioning). Klein [15] has suggested, for example, that expert decision makers engage in a mental simulation that allows them to predict the ramifications of a potential decision prior to taking action.

Overall, the mental model construct has been popular as a means to explain people's understanding of complex systems. The mental model construct has also been useful as a basis upon which to derive hypotheses regarding training strategies for complex systems. A number of studies have been conducted to date; these will be summarized briefly in the following section.

Mental Models and Training

With respect to training, a number of theorists have hypothesized that training that fosters development of accurate mental models of a system will improve performance. According to Rouse and Morris [22], for example, one of the purposes of instruction is to develop mental models necessary to execute the task. Research results regarding mental models and training can be summarized as follows:

- teaching only general principles of system design and function [2, 19]is insufficient; instead, trainees seem to require some form of guidance or cueing in how to apply system knowledge in accomplishing a task [14, 22].
- the manner in which people cognitively structure information about a task has an impact on the way new information is assimilated and learned [7, 28]; new information interacts with existing mental models of the system.
  the impact of pre-existing models of the task can also have a negative impact on training [22]; they can impede learning and may be difficult to eliminate [6].
- the manner in which information is presented has an impact on the formation of initial mental models; that is, people can be led to acquire a particular organization of the material [23, 24, 8].
- allowing people to simply interact with a device or system will often lead to impoverished or incorrect mental models [1, 9].

The implication of findings regarding mental models for training tactical teams will be delineated following a brief review of research into teamwork and team training.

TEAMWORK AND TEAM TRAINING

Despite a considerable amount of research over the past 50 yeasrs, relatively little is known about the nature of teamwork or how best to train teams to perform effectively [3, 4, 11]. In particular, past research has done little to identify specific teamwork skills or investigate how teams acquire, maintain or lose critical teamwork skills. Recently, however, a series of studies conducted with military command and control teams and aircrews has made significant progress in understanding team performance [10, 25]. To summarize the overall findings of this work, the following conclusions can be drawn:

- Behaviors that are related specifically to team functioning (i.e., independent of the particular task at hand) are important to task outcomes [21, 25]
- Effective teamwork behavior appears to be fairly consistent across tasks [21]
- Team process variables (e.g., communication, coordination, compensatory behavior) influence team effectiveness [25]

In terms of specific teamwork behaviors, McIntyre et al. [18] recently reviewed studies of team performance and concluded that teamwork appears to be comprised of a complex of behaviors including: closed-loop communication, compensatory behavior, mutual performance monitoring, giving/receiving feedback, adaptability and coordination. Further, McIntyre et al. suggested that in effective teams, members seem to be able to predict the behavior and needs of other members.

TEAM PERFORMANCE AND MENTAL MODELS

Research cited above provides support for the contention that teamwork behaviors can be isolated from other task-related behaviors. In terms of training requirements and strategies, further research is needed to translate identified teamwork behavioral dimensions into requisite knowledge, skills and abilities (KSAs). For several classes of teamwork behavior such as communication, giving and receiving feedback and mutual performance monitoring, KSA development seems to be fairly straightforward. It is in the area of defining and training skills associated with coordination of action and adaptability that little is known because these skills appear to involve the ability of team members to predict the needs of the task and anticipate the actions of other team members in order to adjust their behavior accordingly.

For example, a study reported by Kleinman and Serfaty [17] investigated the ability of distributed decision-making teams to adapt their behavior to increased workload demands. A significant finding of this work indicated that as workload increased, the team adjusted its strategy so as to affect a trade-off between acceptable performance and sustained workload. Kleinman and Serfaty described two mechanisms by which intra-team coordination changed as workload increased. First, as workload increased to moderate levels, the demand on explicit coordination channels (i.e., where team members coordinate openly via more interactions and sharing of

resources) also increased. However, high workload produced changes in coordination strategies such that constant performance was maintained with a marked reduction in communication. Kleinman and Serfaty [17] interpreted this phenomenon as an "implicit coordination" strategy, where decision makers exercised mutual mental models to anticipate each other's resource needs and actions.

Several other researchers have also suggested that shared mental models may be the basis for effective team functioning. Based on a number of investigations of team behavior in military teams, for example, McIntyre et al. [18] suggested that effective team coordination may be the result of shared mental models of the task. These authors maintained further that effective teams may share mental models of the team as well as of the task. Such a notion may be useful in explaining, for example, the ability of teams to compensate for weaker team members or to distribute responsibility effectively across members.

In other work, Orasanu [20] recently studied the performance of commercial cockpit crews in a simulated emergency scenario. She found that effective aircrews built shared models of the situation that enabled them to manage the emergency. In addition, Whol, Entin, Kleinman & Pattipati, [27] hypothesized that in command and control decision making, a team must have a mutual model of the co-functioning of team members. Finally, in a more extreme position, Klein and Thordsen [16] have introduced the construct of "team mind." These researchers suggest that teams can be conceptualized as a unified information processing unit, analogous in some ways to the individual mind.

In summary, it is clear that the notion of shared mental models has been invoked to help explain complex team behavior, particularly the unique ability of teams to maintain performance in absence of overt communication. In the following sections, the shared mental model hypothesis will be expanded and a discussion of the implications of adopting such an approach for training design will be presented.

<u>Utility of The Mental Model Construct in Teams</u>

The ability of teams to coordinate their actions and adapt to external demands may be best understood in terms of expectations. When a novel situation arises, teams that cannot formulate strategies overtly must anticipate the actions of teammates and demands of the task in order to respond appropriately. The role of mental models in explaining team behavior, then, stems from their ability to allow team members to generate predictions about task and team demands. In fact, the complexity of many team tasks suggests that behavior may be best explained in terms of multiple mental models.

An example may help to illustrate this contention. One of the tasks facing a team of operators in a Navy combat information center (CIC) is to defend the ship against hostile aircraft. Briefly, this task is accomplished by a team who must operate sensor consoles to detect aircraft, integrate and exchange pertinent situation assessment information regarding the aircraft's intent, transmit information to key decision makers, and take action based on the aircraft's believed intent. Typically, such tasks occur under several adverse situational conditions such as high workload, severe time pressure and threat; all conditions that mitigate against explicit coordination strategies.

To be effective in such a situation, a team member must understand the system at several levels. First, he must understand the dynamics and control of the equipment with which he is interacting to extract information. Second, he must understand the task and how to accomplish it (i.e., the significance of information, what information is needed, how information must be combined, and so forth). Third, he must understand his role in the task, that is, what his particular contribution is to the task, how he must interact with other team members, who requires particular classes of information, and so forth. Related to this, he must also know when to monitor his teammates' behavior, when to step in and help a fellow member who is overloaded, and when to change his behavior in response to the needs of the team. Situations of this complexity seem to require, therefore, multiple mental representations of the task: one that describes the equipment, one that describes the task and one that describes the team and his place in it.

Taking this notion one step further, it seems reasonable to hypothesize that the complexity and stability of such models is not equivalent. Specifically, the "equipment" model is likely to be consistent across particular instances of performance; the operator always interacts with the equipment in a similar manner. The "task" model is likely to be more dynamic and complex since a host of situational parameters will vary across task instances and dictate different accomplishment strategies. Still more dynamic is the "team" model which depends not only on the situation, but also on the particular team members involved. In fact, the notion of team adaptability is most clearly understood at this level. Effective teams adjust their strategy to a situation by adopting roles that are most critical to particular task demands and that allow information exchange to be accomplished most efficiently. Implicit coordination (i.e., without communication) can also be explained as a function of mental models of the team, since these allow team members to predict the behavior of teammates and anticipate information requirements in absence of overt strategy formation.

A reasonable hypothesis that stems from the notion of team mental models is that the extent of overlap or commonality among team member mental models will have an impact on team effectiveness. Teams who share common mental models of the task and team are more likely to have accurate expectations regarding the needs of the team, allowing them to adjust their behavior effectively.

Implications of the Team Mental Model Construct

The notion of a shared or team mental model and how it relates to team effectiveness has several implications for the understanding of team performance and training. As an explanatory mechanism, the team mental model construct is useful in understanding how teams are able to coordinate behavior and select task strategies in absence of explicit coordination activities. Under conditions of high workload, time pressure and other kinds of stress, such implicit coordination appears to be crucial [17].

With respect to training, the shared mental model idea suggests that training strategies designed to foster development of shared mental models has the potential to improve team performance. Research cited earlier regarding the success of efforts to train mental models for system operation offers preliminary evidence that such training may be possible. For example, research suggesting that particular knowledge structures (i.e, mental models) can be trained provides support for the notion that common expectations for the task and team can be developed through training.

From what has been presented to this point, it may be hypothesized that specific training strategies which may be useful in training shared mental models include:

1) Positional clarification--interventions designed to provide information regarding the structure of the team and task, the interrelationships among team member positions, and the roles and responsibilities of each team member could be hypothesized to improve team performance by enhancing common task and team expectations. Such training, which represents requisite team and task knowledge, could be presented via lecture, computer assisted instruction, or via written material. Such training would represent initial preparatory training, but would probably not be sufficient to develop shared mental models. Another potential training technique that may be useful for this purpose is role playing, which also has the benefit of making the trainees more active participants in the training.

2) Guided practice and feedback--Results cited above showing that unguided practice can lead to inaccurate mental models suggests that teams should practice tasks under the guidance of instructors. Feedback and debrief mechanisms must be designed to result in accurate, common expectations for the task and team. In addition, feedback regarding specific behaviors that must be changed should be more effective in establishing accurate expectations than general, less specified feedback. Simulation facilities would be a most appropriate means to provide such practice opportunities. Recent evidence with aircrews suggests that low-fidelity simulation may also be viable (Stout et al., 1990).

3) Cross training--A potentially useful strategy to train common mental models may be to cross train team members on tasks that are related to their own task. Such training would be beneficial to the extent that it helps team members to learn what their teammates will need (in terms of resources, information, assistance) given various task demands.

4) Instructor training--Much of the success of team tactical training depends on the quality of instructors. With respect to shared mental models, instructors must be trained to recognize effective teamwork behaviors and other evidence that team members share common mental models as a basis to deliver feedback.

5) Team leader training--Training team leaders to foster development of shared mental models also has potential value. It can be hypothesized that team leaders who are trained to articulate their own view of the task and team, who encourage discussion and strategy formation among team members, and who make clear their expectations of team member behavior should be successful in helping their teams to develop shared mental models.

SUMMARY

It has been argued in this paper that the notion of shared mental models in teams may have value as a means to understand effective team performance and as a basis to develop team training strategies. Particularly with respect to adaptability and coordination, the mental model construct helps to explain how teams are able to perform effectively in absence of overt or explicit avenues of strategy formation. However, several areas of research are necessary if the shared mental model hypothesis is to be useful. First, methods to measure team mental models must be developed. Second, a means to diagnose and compare mental models across team members must be devised so that their relationship to effectiveness can be determined. Finally, training strategies that will have an impact on shared model development (such as those listed above) must be established. Based on evidence regarding mental models in system control and the nature of team performance, such investigation may provide crucial data regarding how to train teams to perform optimally.

REFERENCES

[1] Bayman, P. & Mayer, R. E. (1984). Instructional manipulation of users' mental models for electronic calculators. International Journal of Man-Machine Studies, 20, 189-199.

[2] Brigham, F., & Laios, L. (1974). Operator performance in the control of a laboratory processing plant. Ergonomics, 18, 53-66.

[3] Cummings, T. G. (1981). Designing effective work groups. In P. C. Nystrom & W. Starbuck (Eds.) Handbook of Organizational Design (Vol. 2). London: Oxford University Press.

[4] Cannon-Bowers, J. A., Oser, R., & Flanagan, D. L. (1990). Work Teams in Industry: A Selected Review and Proposed Framework. To appear in R. W. Swezey & E. Salas (Eds.), Teams: Their Training and Performance, New York: ABLEX.

[5] Cannon-Bowers, J. A., Salas, E. & Converse, S. A. (1990). Cognitive psychology and team training: Training shared mental models of complex systems. Human Factors Bulletin. Santa Monica, CA: The Human Factors Society.

[6] DiSessa, A. A. (1982). Unlearning Aristotelian physics: A study of knowledge-based learning. Cognitive Science, 6, 37-75.

[7] Eberts, R. E., & Brock, J. F. (1987). Computer-assisted and computer-managed instruction. In G. Salvendy (Ed.), Handbook of Human Factors. New York: John Wiley and Sons.

[8] Eylon, B. S., & Reif, F. (1984). Effects of knowledge organization on task performance. Cognition and Instruction, 1, 5-44.

[9] Frederiksen, J. & White, B. (1989). An approach to training based upon principled task decomposition. Acta Psychologia, 71, 89-146.

[10] Glickman, A. S., Zimmer, S., Montero, R. C., Guerette, P. J., Campbell, W., J., Morgan, B. B., Jr., & Salas, E. (1987). The Evolution of Teamwork Skills: An Empirical Assessment with Implications for Training. (Technical Report No. 87-016). Orlando, FL: Naval Training Systems Center.

[11] Hackman J. R., and Morris, C. G. (1975). Group tasks, group interaction process and group performance effectiveness: A review and proposed integration. In L. Berkowitz (Ed.), Advances in Experimental Social Psychology (Vol. 8). New York: Academic Press.

[12] Jagacinski, R. J., & Miller, R. A. (1978). Describing the human operator's internal model of a dynamic system. Human Factors, 20, 425-433.

[13] Johnson-Laird, P. (1983). Mental Models. Cambridge, MA: Harvard University Press.

[14] Kieras, D. E., & Bovair, S. (1984). The role of a mental model in learning to control a device. Cognitive Science, 8, 255-273.

[15] Klein, G. A. (1989). Recognition-primed decisions. In W. B. Rouse (Ed.), Advances in Man-Machine Systems Research, 5, 47-92. Greenwich, CT: JAI Press.

[16] Klein, G., & Thordsen, M. (1989). Recognitional decision making in C2 organizations. Paper presented at the Symposium on C2 Research, National Defense University, Washington, D.C., June, 1989.

[17] Kleinman, D. L., & Serfaty, D. (1989). Team performance assessment in distributed decisionmaking. In R. Gilson, J. P. Kincaid, & B. Goldiez (Eds.), Proceedings: Interactive Networked Simulation for Training Conference, Orlando, Florida, April, 1989.

[18] McIntyre, R. M., Morgan, B. B., Jr., Salas, E., & Glickman, A. S. (1988). Team research in the eighties: Lessons learned. Unpublished manuscript, Naval Training Systems Center, Oralndo, FL.

[19] Morris, N. M., & Rouse, W. B. (1985). The effects of type of knowledge upon human problem solving in a process control task. IEEE Transactions on Systems, Man and Cybernetics, SMC- 15, 698-707.

[20] Orasanu, J. (1990). Shared mental models and crew performance. In Proceedings of the 34th Annual Meeting of the Human Factors Society, Orlando, Fl, October, 1990.

[21] Oser, R., McCallum, G. A., Salas, E., & Morgan, B. B., Jr. (1989). Toward a definition of teamwork: An analysis of critical team behaviors. Technical Report No. TR-89-004, Naval Training Systems Center, Orlando, FL.

[22] Rouse, W. B., & Morris, N. M. (1986). On looking into the black box: Prospects and limits in the search for mental models. Psychological Bulletin, 100, 349-363.

[23] Shavelson, R. J. (1972). Some aspects of correspondence between content structure and cognitive structure in physics instruction. Journal of Educational Psychology, 63, 225-234.

[24] Shavelson, R. J. (1974). Methods for examining representations of a subject-matter structure in a student's memory. Journal of Research in Science Teaching, 11, 231-249

[25] Stout, R., Cannon-Bowers, J. A., Salas, E., & Morgan, B. B., Jr. (1990). Does crew coordination behavior impact performance? In Proceedings of the 34th Annual Meeting of the Human Factors Society, Orlando, Fl, October, 1990. 20

[26] Veldhuyzen, W., & Stassen, H. G. (1977). The internal model concept: An application to modeling human control of large ships. Human Factors, 19, 367-380.

[27] Whol, J. G., Entin, E. E., Kleinman, D. L., & Pattipati, K. (1989). Human decision processes in military command and control. In W. B. Rouse (Ed.), Advances in Man-Machine Systems Research (Vol. 1). Greenwich, CT: JAI Press.

[28] Wickens, C. D. (1984). Engineering Psychology and Human Performance. Columbus, OH: Merrill.

[29] Williams, M. D., Hollan, J. D., & Stevens, A. L. (1983). Human reasoning about simple physical systems. In D. Gentner & A. L. Stevens (Eds.), Mental Models, Hillsdale, NJ: Erlbaum.

ABOUT THE AUTHORS

Dr. Eduardo Salas is a Senior Research Psychologist in the Human Factors Division of the Naval Training Systems Center. He holds an M.S. in Industrial Psychology from the University of Central Florida and Ph.D. in Industrial/Organizational Psychology from Old Dominion University, Norfolk, VA. He is principle investigator for NTSC's R&D program on team training and performance, and project manager of an effort to develop aircrew coordination training for Navy and Marine pilots. Dr. Salas has conducted research in team and individual training design, training evaluation, job/task analysis, skill acquisition and personnel psychology. His most recent project responsibilities are as manager of an effort to develop and validate aircrew coordination training for the Navy and Marine Corps and laboratory director for the Tactical Decision Making Under Stress (TADMUS) project.

Dr. Janis A. Cannon-Bowers is a Research Psychologist in the Human Factors Division of the Naval Training Systems Center. She holds an M.A. and Ph.D. in Industrial/Organizational Psychology from the University of South Florida, Tampa, FL. She has been involved in a number of research projects, including specification of training needs and design for aircrew coordination training, assessment of team training needs and investigation of transfer of training issues. Her research interests include team training and performance, crew coordination training, training effectiveness, and tactical decision making. Currently, Dr. Cannon-Bowers is co-principle investigator for the Tactical Decision Making Under Stress (TADMUS) project, conducting research concerned with improving individual and team decision making in the Navy tactical environment.

Ms. Catherine V. Baker is a Post-Graduate Fellow in the Human Factors Division of the Naval Training Systems Center. She currently holds an M. A. and is completing work on her Ph.D. in Industrial/Organizational Psychology at the University of South Florida, Tampa, FL. She is responsible for conducting behavioral research in support of NTSC's TADMUS program. This involves conceptualizing and planning research projects relevant to issues in team training and performance. Her research interests include crew coordination training, tactical decision making, and the development of training and simulation principles to mitigate the effects of stress on team decision making processes.

INSTRUCTIONAL DISPLAY DESIGN FOR SUBMARINE TACTICS TRAINING

Dr. Thomas J. Hammell
Paradigm Associates
East Lyme, CT
and
Dr. Robert H. Ahlers
Naval Training Systems Center
Orlando, FL

ABSTRACT

Instructional features, such as performance measurement and feedback, are widely accepted as integral elements of instructional systems, although their effectiveness is often compromised by inadequate use. This investigation focused on the design of feedback displays used to convey information to students during debriefing following a team tactics training exercise. Such feedback displays are essential components of tactics training but must be designed to motivate use. Operational system displays and capabilities, which are familiar to instructors and students, were evaluated to determine if they could provide the basic mental model foundation on which to build instructional enhancements. Feedback displays designed in accordance with instructor and student operational system mental models were found to facilitate user acceptance and ease of use. The results of questionnaires administered to, and information retrieval tasks performed by, twenty-one active duty submarine tactics instructors showed strong approval of the mental model feedback display design approach and superior information processing performance.

INTRODUCTION

Instructional technology features are today widely accepted as integral parts of a computer-based training system, crucial to the enhancement of training effectiveness. Examples include system-generated measures of effectiveness (MOEs) and performance feedback displays for identifying student performance strengths and weaknesses. These features are designed for use primarily during post-scenario debriefings.

The effectiveness of instructional features is dependent on several factors, including their regular and proper use by instructors. The design of instructional features should encourage regular use as well as provide effective instructional assistance. Their design should be tailored for the instructor and students with the same care espoused for user-interface design in operational systems, albeit following principles appropriate to the design and operation of instructional systems.

Recent research suggests the relevancy of a mental model concept to computer display design, workplace design, and to the learning/training process.[2,3,5,6,4] The preponderance of decision making tasks in military operations, such as submarine tactics, suggests the importance and existence of strong cognitive mental models by operational personnel. These factors are being carefully considered in the design of operational combat system human interfaces.[1] Training systems should likewise carefully consider the use of operational personnel (instructors and students) mental models in the design of instructional features.

It was with the goals of improving the usability and acceptance of instructional technology features, specifically feedback displays, that this study was conducted. The study hypothesized that feedback displays should be designed in accordance with instructor and student operational system mental models to facilitate user acceptance and ease of use. A mental model, for purposes of this study, can be viewed as an individual's internal cognitive representation of a physical system and its functioning. Instructional enhancements should be added to the operational displays in a form as close as possible to that of operational system/display information and features, so as to maintain a good mental model. User acceptance and usability will be high to the extent that the resultant feedback display exhibits information characteristics similar to those of the operational system. Where instructional features are unique to the trainer, they should be modeled as close as possible to good operational system characteristics.

STUDY OBJECTIVE

The objective of the study was to evaluate an approach to the design of instructional displays, specifically feedback displays, that focuses on design in accordance with instructor and student operational system mental models. This approach further relies on the adaptation of operational system displays, as the mental model framework, for the initial foundation onto which the instructional feature enhancements should be built. Certain operational system displays are believed to represent fundamental mental model

narrow menus along each of the four display margins. An upper horizontal display menu was suggested for presentation of a scenario time bar, enabling the instructor to conveniently move to desired time points during the debriefing replay. A narrow vertical menu on the left side was suggested for display selection, while that on the right was suggested for overlay selection.

Instructional enhancements to the displays included:

1) Actual target-related graphical information was overlaid on geographic and other plot windows. For example, the actual target track (as known by the simulation) was overlaid on the geographic plot, along with the combat system's target track (as known by the students).

2) Actual target-related alphanumeric information was presented next to the corresponding system-generated information. Examples include target bearing, range, course, and speed.

3) MOEs were displayed. These included MOEs to be available on the operational combat system (e.g., probability of counterdetection), and MOEs available only in the trainer (e.g., target range, course and speed error).

4) Windows and specific overlays of relevant combat system-generated tactical information, such as a target escape envelope, were used.

5) Theoretical doctrine values, such as for weapon presets, were presented adjacent to the normally displayed list of system-generated presets.

6) Projected outcome situations, both graphical and alphanumeric, were provided so they could be compared with those actually achieved. For example, weapon search coverage was projected at an early stage prior to firing; its actual coverage was shown as a weapon deployed.

7) Historical system-generated data distributions were displayed along with the actual target information. For example, the system-generated target range error estimates formed an envelope over time, illustrating the changing nature of the TMA solution accuracy as a function of the situation, target and ownship actions.

8) Several feedback display control functions were provided to facilitate instructor use during the debriefing. These included functions common to a windows/mouse interface, as well as functions more unique to the debriefing application. Examples are enlarged windows to permit focusing on particular graphical information, selection and removal of overlays, rapid jumping between different displays, and rapid jumping to time points in the scenario.

An illustration of the adaptation of a hypothetical operational display for instructional purposes is shown in Figure 1. Instructional features include the actual target track history (in bold) and current position overlaid on the combat system-generated track history (thinner lines); comparison of the system-generated and actual target parameters in the data area on the right; measures of effectiveness in the same area; and various mouse-activated features. A horizontal slider control is included at the top to select replay time position in the scenario, instructional overlay selection icons are placed on the right vertical margin, and icons for selecting another display are located on the left margin. These modifications were made to an operational display as necessary to support the instructional process (in this case, post scenario debriefing).

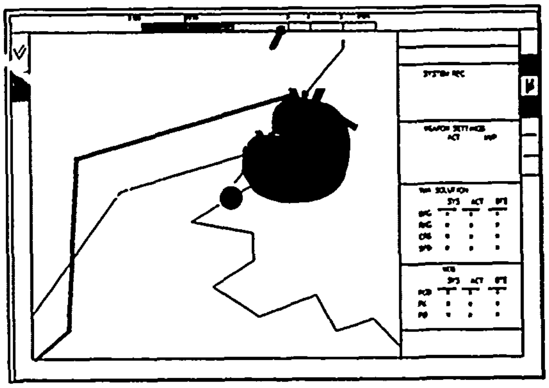


Figure 1. An illustration of a hypothetical operational display adapted for instructional purposes.

Subjects

Twenty-two active duty submarine tactics instructors at three training sites, including Atlantic and Pacific fleets, served as subjects.

The instructors/subjects were not directly familiar with the operational displays since the advanced combat system is still under development. However, they were familiar with many of the individual elements contained in the displays and with the general format of the information.

Apparatus

The two new feedback displays were generated by a drawing program in color on

structures of operational personnel, the population from whom the instructors and students come.

APPROACH

The empirical research was conducted in the context of tactical team training for an advanced submarine combat system, with two newly designed feedback displays to support the post-scenario debriefing session. Two submarine combat system operational displays, representing good submarine tactics mental models, were selected as the foundation for the two new feedback displays. Instructional enhancements were added to the operational displays to create the new feedback displays designed for use in the context of post-scenario team training debriefings.

Evaluation of the design hypothesis was achieved using two types of empirical data: 1) questionnaire data investigating instructor acceptance of the new displays and the hypothesized display design approach; and 2) reaction time and accuracy data measuring instructor information retrieval performance comparing the newly designed displays with currently available feedback displays.

Each of the two new feedback displays was presented as a series of photographs, representing sequential time slices and information overlay options, during a hypothetical post-scenario debriefing. Each subject answered tactical questions using information presented on each of the new displays, and on corresponding versions of the currently available feedback displays. Response time and accuracy data were collected. Each subject answered a detailed questionnaire immediately following the display presentation.

Display Selection

Tacticians, including instructors and experienced students, were hypothesized to develop mental models corresponding to the graphical operational combat system displays with which they regularly work. Displays of this type, therefore, were hypothesized to provide good baseline designs, the foundation platforms, to which necessary instructional information could be added to achieve effective and user friendly instructional displays. This approach used operational displays selected for conformance to the experimenter-perceived instructor and student mental models, in addition to other criteria, as the foundation onto which the instructional displays would be built.

Instructional features, such as the actual target track history overlaid on the combat system generated target track history, were added to a simulation of an operational display to achieve an instructional display which was then suitable for debriefing purposes. Elements from multiple operational displays were integrated into the simulated instructional display as appropriate for training.

Two operational combat system displays two operational combat system displays were selected from twenty-five candidates, which had been identified on the basis of providing a comprehensive view of the combat situation. Selection of the two operational displays was accomplished on the basis of: 1) tactics training application, for which an instructional process strategy was hypothesized; and 2) specific display selection criteria, described below. A comprehensive submarine tactics training strategy was hypothesized for the next generation combat system. A subset of the strategy was assumed for this experimental application, constraining the set of candidate displays.

Each of the twenty-five displays was assessed on a 3-level Likert-type scale for each of nineteen characteristics. These criteria addressed the potential of an operational display to be adapted for instructional feedback. The information content, relevancy, and presentation to provide an image of the tactical situation (in essence, a good mental model) was also addressed. The displays were then rank-ordered on the basis of their assessment score, and the highest ranking display in each of two tactical application areas was selected.

A Weapons Status display, showing a geographical picture of the engagement and associated alphanumeric information, with overlays in both the geographic and alphanumeric areas, was selected as providing a comprehensive overview of the tactical situation. It provides a good summary of the overall tactical situation in a form familiar to submarine tacticians. A Target Motion Analysis (TMA) Summary display was also selected. It presents a mosaic of TMA-relevant graphical information windows and an alphanumeric window. This TMA display was believed to provide a more in-depth set of information in one particular area of submarine tactics, target motion analysis. These two operational displays formed the foundation for development of the new feedback displays.

Feedback Display Design

Instructional enhancements were integrated with the operational displays. These enhancements were of two types: 1) other characteristics of the operational combat system which have relevancy to the instructional process, and 2) features which are indigenous to the training situation only. These consisted of actual target and situation information, transformations of selected tactical information, and manipulation of presented information (e.g., enlarging windows) that were presented as overlays for comparison with the similar information generated by the combat system. Variations of enhancements were presented during the evaluation of each display.

The instructor control interface, which is of direct importance to user acceptance, was considered in the design of the feedback displays, although to a lessor degree. A mouse-type interface was suggested, with

a PC/AT computer display system. Approximately twenty display frames were developed for each display. These represented time slices and overlay options appropriate to the debriefing sessions of the respective hypothetical tactical training exercises. The resultant series of display frames were printed on 5"x8" photographs for serial presentation.

In addition to the two modified operational displays, feedback displays from a current submarine combat system trainer were evaluated. These displays (called "trigraphs") simultaneously display three linegraphs of selected parameters (y) as a function of scenario time (x). Each trigraph presented three parameters, for a total of nine curves.

Three trigraph displays were developed to correspond with the three new feedback display frames (two different time-slice versions of a Weapons Status display and one TMA display). The new feedback displays and corresponding trigraphs were the display frames used for the information retrieval tasks, directly comparing the new and current designs.

A questionnaire was developed to obtain subjects' opinions regarding the design approach, quality, and content of each display. Each questionnaire included questions with a five-category Likert-type response (31 and 25 questions for the Weapons Status and TMA displays, respectively), and open-ended comments (8 questions each). Additionally, the TMA display questionnaire included questions of a more general nature pertaining to the design and use of training systems.

A set of tactical questions was developed for each of the three trigraph-new display combinations (8 or 9 questions), comprising the information retrieval tasks. All of these questions were tactically relevant for the situation, and would be expected to occur during a normal debriefing.

Procedure

The data collection sessions were individually conducted with each subject, spanning a period of 1 1/2 to 2 hours each. Each subject, at the start of each session, received a brief overview of the advanced combat system capabilities and the two operational displays which formed the foundation of the new feedback displays. The subsequent steps are explained below.

Information Retrieval. Each subject initially performed the information retrieval task. A photograph of a Weapons Status, TMA, or trigraph feedback display was presented and its content explained. After studying the photograph for about a minute, the series of tactical questions was asked, one at a time, with responses and times recorded by the experimenter. This information retrieval process was carried out for two pairs of displays, with the presentation order first alternating between the Weapons Status and corresponding trigraph displays, and then the TMA and its corresponding trigraph display. The Weapons Status/trigraph combination was presented first. This would be expected to occur in a training situation because the Weapons Status display provides a tactical situation overview, while the TMA display provides a more detailed view of a tactical subset. The order of new and corresponding trigraph displays was balanced across subjects.

Instructor Acceptance. A submarine tactical training scenario engagement, with assumed team actions and outcome, was summarized for each instructor. It was immediately followed by a debriefing session using the new feedback displays, as might be conducted by an instructor. The debriefing, explained by the experimenter, was accomplished using the series of about twenty feedback display photographs. They simulated a replay of the scenario, along with providing insight into the various instructional technology enhancements available. The questionnaire was administered immediately following completion of the debriefing. This process was first conducted for the Weapons Status display, and then for the TMA display in the context of a different training exercise.

FINDINGS

The analysis addresses the findings separately below for user acceptance of the displays (questionnaire data), and information processing performance (information retrieval).

User Acceptance of Feedback Displays

Of greatest importance to the study was the acceptance of the new feedback displays by the experienced instructors. Six questionnaire items directly pertain to this issue:

A. Rate the overall quality of this display for assisting the instructor in presenting the debriefing.

B. How does this type of display compare with the traditional debriefing methods for providing feedback?

C. Rate this type of display for the ease of recognizing and understanding information.

D. How often would you expect to use this display, or a similarly designed display, during the debriefing?

E. Rate the effectiveness of designing debriefing displays based on operational system displays, such as these. (TMA questionnaire only)

F. Will this display design approach foster its use during the debriefing? (TMA questionnaire only)

The opinions of the experienced submarine tactical instructors on these questions are summarized in Figure 2. A score of one indicates a negative response; a five is

highly positive; three indicates a neutral response. The consistently high user acceptance of these feedback displays shows the efficacy of using a good operational display foundation. Of substantially greater interest, these results strongly support the importance of designing feedback displays in accordance with instructor/student mental models, the hypothesized design approach investigated by this study. The inherent mental model characteristics of good operational displays facilitate the design of effective feedback displays.

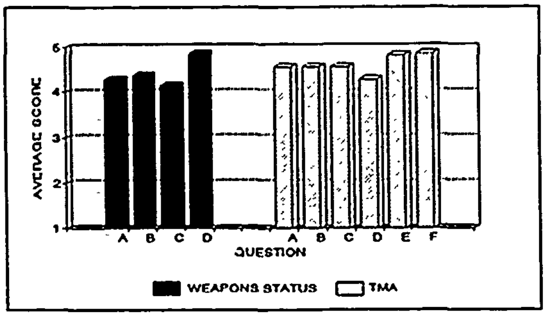


Figure 2. Average opinion ratings of new display formats (5 is highest possible rating).

The questionnaire responses provided additional information supporting this design approach, and a variety of other information addressing specific characteristics of the two newly designed feedback displays. These included characteristics that were rated high and others that were rated low. The overall feedback display design approach, however, was viewed very positively by the operational submarine tactics instructors.

<u>Information Processing Performance</u>

The information processing performance achieved using the newly designed feedback displays is of considerable importance in the applied training environment. A display design, even a generic design approach, might be ineffective even though it shows a high degree of user acceptance. If the design approach of exploiting already formed mental models is valid, however, information processing performance would be expected to correlate with display acceptability.

<u>Response Accuracy</u>. The subjects responded to the tactical questions with significantly higher accuracy for the newly designed feedback displays, in comparison to the currently available trigraph displays (see Figure 3). While using the newly designed feedback displays instructors achieved greater accuracy on 24 of 28 responses (86%), less on only 2 (7%), and the same on 2. This result would be expected to occur by chance in only 1 of 10,000 similar tests ($p<.0001$, non-parametric Sign Test), a highly significant finding.

The tactical questions posed to the instructors were comprised of: 1) procedural questions that required the rapid location of alphanumeric information on the display; and 2) interpretive questions for which the answer was not immediately available, requiring analysis of displayed information. Although the interpretive questions imposed a greater cognitive burden on the subjects, both interpretive and procedural questions resulted in significantly better performance using the new displays (see Figure 3). On interpretive questions, instructors achieved significantly higher accuracy on 11 of 15 responses (73%) while using the newly designed displays, less on 2 (13%), and the same on 2 ($p<.02$). Instructors were more accurate on all 13 procedural responses (100%)($p<.0004$, Sign Test).

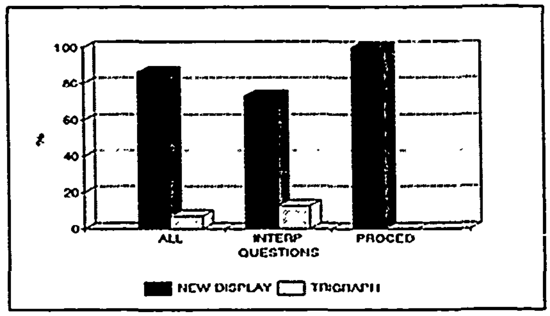


Figure 3. Percentage of questions that resulted in a greater number of correct answers.

<u>Reaction Time</u>. The times taken to answer each question were similarly found to differ between the new display designs and the currently available trigraph displays (see Figure 4). Note that two of the questions had two responses each; hence, there were more accuracy responses than reaction times. The average reaction times to answer the tactical questions using the newly designed displays were faster on 21 of the 26 questions (81%), less on 2 (12%), and the same on 2 ($p<.0004$, Sign Test). Also similar to the response accuracy findings, the reaction times were significantly faster on interpretive questions when using the new displays, with 12 of 13 questions yielding faster average responses (92%), and slower on only 1 question (8%) ($p<.003$, Sign Test); and on the procedural questions, with 9 of 13 questions yielding faster average responses (69%), slower on 2 questions (15%), and the same on 2 questions ($p<.03$, Sign Test).

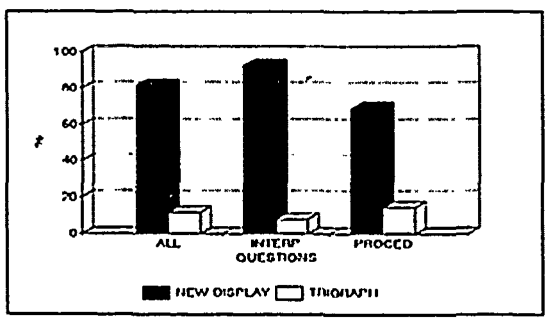


Figure 4. Percentage of questions that resulted in faster reaction times.

The response accuracy and reaction time results strongly demonstrate superior information processing characteristics of the Weapons Status and TMA displays, in comparison with the trigraph displays currently on the submarine combat system trainers. Subjects were better able to locate, retrieve, and process information from those displays which were more similar to their own mental models. Although the newly designed displays are certainly less familiar to the instructors than the trigraph displays which exist on the trainers, elements of these new displays are also obviously more familiar to the instructors -- more like their mental models of the submarine tactical engagement.

CONCLUSIONS

The user acceptance and performance results consistently support the effectiveness of display design approach used to achieve the new feedback displays. Carefully chosen operational displays can provide a good mental model foundation on which to build instructional assistance displays. The subjects' enthusiastic acceptance of the newly designed feedback displays, coupled with superior performance using these unfamiliar displays, demonstrate the importance of designing information displays in accordance with the user's mental model, not only to achieve user acceptance, but also to achieve enhanced performance.

The methodology used in this study was conducted in a submarine tactics context. The principles employed, however, should apply in many application contexts. The specific aspects of instructional technology used, and the resultant display/control designs would need to tailored to the particular application.

The use of a mental-model-based display design approach is believed particularly important in instructional systems since the instructors and students do not generally have much time for the learning of instructional system displays and their operation. Hence, the information presented on displays for feedback should be in an easily recognized and usable form for instructors and students -- as similar as possible to the operational system mental models they already possess.

REFERENCES

[1] Kanter, M. F., O'Brien, F. J. "Submarine Combat System Configuration Modeling Methodology". Technical Memorandum TM No. 89-2032, Naval Underwater Systems Center, January 3, 1989.

[2] Norman, D. A., and Babrow, D. G. "On the Role of Active Memory Processes in Perception and Cognition". In "The Structure of Human Memory", Cofer (Ed.). W. H. Freeman, San Francisco, 1976.

[3] Rasmussen,J. "Information Processing and Human-Machine Interaction". Amsterdam: North-Holland, 1986.

[4] Veldhuyzen, W., and Stassen, H. G. "The Internal Model: What Does it Mean in Human Control?" In T. Sheridan and G. Johannsen, (Eds.), "Monitoring Behavior and Supervisory Control". Plenum, New York, 1976.

[5] Williams, K. E., Hamel, C. J., and Shrestha, L. B. "An Evaluation of Characteristics contributing Towards Ease of User-Computer Interface in a Computer-Aided Instruction Exercise". Technical Report TR87-030, Naval Training systems Center, Orlando, FL, 1987.

[6] Wilson, J. R., and Rutherford, A. "Mental Models: Theory and Application in Human Factors". The Journal of the Human Factors Society, 31(6), 617-634, 1989.

ABOUT THE AUTHORS

DR. THOMAS J. HAMMELL is President and Chief Scientist of Paradigm Associates, a small business engaged in human factors work for the marine and education industries. He has had extensive experience conducting research and development programs for the Navy, Maritime Administration, Coast Guard, Nuclear Regulatory Commission, and private industry, research on advanced instructional concepts and systems, port and waterway development, the human-computer interface, command and control decision making, maintenance manning, naval tactics, and the certification of mariner training and licensing systems. He has been a member of National Academy of Science panels, and an adjunct faculty member at the University of Connecticut, Engineering and Psychology Departments. Dr. Hammell's education consists of Bachelor of Engineering and Master of Management Science degrees from the Stevens Institute of Technology, and the Ph.D. in experimental psychology from the University of Connecticut. He has authored about 100 publications in his professional field.

DR. ROBERT H. AHLERS is a Research Psychologist with the Human Factors Division of the Naval Training Systems Center. He has managed research projects concerned with the application of knowledge-based modeling to the simulation of intelligent aspects within a training environment. His recent work addresses the use of artificial intelligence techniques to provide certain instructional functions, such as performance diagnosis, and to provide tactical adversary capabilities. He graduated from the University of Virginia with B.A. and M.A degrees in experimental psychology and from North Carolina State with a Ph.D. in Human Factors.

Tactics as Decision Making: Issues in Tactical Training Development

Jerome Bresee and Michael Naber

Delex Systems, Inc.
Vienna, VA

ABSTRACT

Military training system designs are typically optimized for the demonstration and practice of operational procedures, and seldom focus specifically on tactical decision making. The normal approach for training systems design consistently suggests the maximum use of physical fidelity, while leaving the user to decide how to make use of that fidelity. This is usually done by training the performance of tactical procedures in the environment that was used for the training of operating procedures.

This paper is based on the premise that training in tactical decision making has certain fundamental differences from procedural training, and therefore different requirements for training strategies and media. The paper offers some observations on the tasks that make up tactical decision making behavior, and identifies some related training requirements. A set of guidelines for implementing these requirements is offered, followed by a description of some suggested training environments.

Positions advanced in the paper are supported by experience gained through the development of tactical instruction and tactics-oriented training devices in naval submarine, surface and air warfare.

INTRODUCTION

During 1944, the Fourth Fighter Group of the 8th Air Force contained within its ranks a remarkable number of highly effective combat pilots. Two of them are a study in contrasts which show the end result of what could have been two approaches to training.

The Fourth Fighter Group was commanded by Col. Donald Blakeslee, a former RAF Eagle Squadron commander, who was certainly one of the best air combat leaders of his era. It is reported that Blakeslee was able to keep track of a massive air battle in great detail, gathering and assessing information rapidly enough to amaze his pilots with specific instructions in the middle of the fight. Blakeslee was a tactician with skill enough to use fifty aircraft as a weapon in real time. However, Blakeslee himself "couldn't shoot worth a damn," and perhaps was not as polished in his procedural skills as many of the men he commanded.

Don Gentile was of another sort. A polished pilot and excellent marksman, he pursued an individual opponent with intense focus and, in air-to-air kills, was the high scorer of the group. A story appearing in an informal history of the group written by their public relations officer describes one engagement in which Gentile pursued three opponents to the limit of his altitude and fuel, transmitting eventually, "Tell 'em I got two if I don't get back." Gentile was a great individual tactician also, but of a much narrower scope than Blakeslee.

Obviously, it would be a wonderful thing to turn out people with the strengths of both these men; people who could perform as well tactically as Blakeslee, and who could fly and fight as well as Gentile. Usually, though, the capabilities of a Don Blakeslee come much later in a career. Procedural and psychomotor skills are the usual initial product of our training systems, with broader tactical skill and judgement coming later. This paper explores in a limited way the nature of tactical skill, and offers some thoughts about a practical sequence of training experiences to help achieve this skill more quickly.

TACTICAL TRAINING AND PROCEDURAL TRAINING

The easiest way to get someone to do something for the first time is to tell them exactly what to do, step by step. Never mind why, or even when, just do what you're told, when you're told to do it. This approach gets the student doing something. And, provided instructions are understood and carried out, it gets the student doing something correctly. This may be why so much of military instruction is procedural.

By consequence or by design, most military jobs have a great many highly procedural components. There are checklists for each evolution in an aircraft. There are troubleshooting trees in maintenance, with a step by step series of tests, measurements, and corrective actions. And there are usually a series of procedures associated with the employment of weapons and other strictly tactical actions. Procedural training is a time-proven technique for developing a great many military job skills.

An expert performer of a procedural tasks achieves that level through repetition, and procedural training design focuses on

this fact. Training for procedural skills is learning through repetition to generate the same actions over and over again, with great reliability. After many repetitions, the trainee gets into a groove, and can produce the same results every time.

Practice is probably the only certain way to improve the performance of any task. It has long been known in public education that the most effective learning improvement step you can take is to increase the number of opportunities for practice with feedback. But training a tactical decision maker is a different challenge. The tactical decision maker will never see the identical situation twice, and in any case must avoid being predictable. Tactically, getting into a groove can be dangerous.

This is the way in which tactical behavior is different from operator behavior. Operator behavior is procedural, limited to the functions that a machine or system can perform. While proficient operator behavior is essential to successful tactics, proficient operator behavior is not enough for successful tactics. Tactical behavior is the generation of a solution to a specific problem never before encountered.

TACTICAL TRAINING REQUIREMENTS

If tactical behavior really is different from procedural behavior, it stands to reason that it must have different training requirements. If this is the case, then these requirements ought to be identifiable. A top-level task analysis of tactical decision making process might yield the following behaviors (adapted from Van Gundy, 1981):

1. Gather and analyze problem information
2. Generate alternative problem definitions
3. State the problem
4. Identify classes of appropriate actions
5. Define areas of uncertainty
6. Generate possible alternative solutions
7. Search for/gather information to evaluate solutions
8. Generate solution consequences
9. Select a tentative solution
10. Implement first action toward solution
11. Evaluate results
12. Select follow-on action

Tactical decision-making is problem solving behavior, performed under high stress. Given large amounts of data, only some of which is relevant, all of which is dependant upon the quality/fidelity of the observer/sensor, perhaps out of date, and finally, little time in which to act (the tactician in battle cannot refer to a committee for a consensus), a course of action must be chosen. Each situation is unique, and a unique solution must be developed. If this was all there was to it, good tactical performance could only be achieved randomly.

A good tactician does not behave randomly. Faced with a situation where only a few firm rules exist and all else can change as a result of time passing or unknown actions initiated by the enemy, the good tactician can identify all the options realistically available and select the best of them. When this performance is examined after the fact the choice that was made seems obvious. The real trick seems to lie not so much in the choosing, but in identifying the available choices. Therefore, among other things, good tactical training ought to help students to generate accurate and realistic sets of alternatives from which to choose.

Naturally enough, the premise that seems to drive much tactical thinking is the limitation of choices. In airborne anti-submarine warfare, you can place a sonobuoy anywhere on the ocean surface you want. You are limited only by the fuel you carry. This is only a two-dimensional field, and already the set of alternatives is infinitely large. Clearly, you need to make this set more manageable. Therefore, the first thing that you are usually taught is identification of the constraints that apply.

Tactical options are ultimately constrained by physical limits. These limits certainly start with the mission. The tactician's intention defines the outer boundaries of the available options. The tactician next considers the capabilities of the available weapons, sensors, and of the platform, and the total set of capabilities these factors define. The adversary's capabilities can either constrain the tactician further or open up more options, as can the physical environment. This may involve the maximum speed of a ship, the physics of sound in seawater, the thickness of a cloud deck, or whether or not a recent rain has turned the roads to mud.

Constraints limit alternatives, despite extraneous variables. The tactician uses these constraints to define the broad classes of options available. These will run from the most practical to the barely possible. He can evaluate these options on a range of criteria, combine his evaluations into a chosen course of action, and do so quickly. The tactical training environment must allow the student to acquire and make use of this data so that certain classes of solutions become discernable. As this happens, it becomes clear that systematic applications of a broad-based techniques are feasible, and a practical set of choices can be defined

Learning to identify these broad-based techniques, together with developing skill in applying them means that the student needs lots of opportunities for making decisions, or lots of different practice examples. Identifying the possibilities and picking the best course of action can only be done once at any given point in

time. "Trying again," or repeating a problem that has been solved before turns training in decision making into training in remembering old decisions. The student must gain experience in analyzing new data and testing tactical guidelines against each new problem. If the problems are realistic, the problems themselves will show the student the real usefulness of basic tactical techniques.

To make the point another way, while there are an infinite number of variations in any situation, the capabilities and limits of weapons, platforms, sensors, threat and crew, along with the limits imposed by weather and other environmental factors, effectively limit the number of general approaches available. If a squad of infantry must occupy a certain position on the other side of a hill, they can either go over the hill or around it. The precise route selected for either option may make a world of difference, but the choice between climbing and detouring colors everything else.

Controlling the nature and accessibility of information is a key requirement of a tactical training environment. Based on information available, the trainee identifies options and chooses one. Once the choice is made, information about the results of the choice is essential, both during the problem and at its end. Most tactical scenarios involve a series of choices. Once sufficient data is collected and assessed in light of the mission, the tactician must select an opening gambit, assess its results, and make the next decision. For example, if the weapon available is a stand-off missile, this opening gambit may be a maneuver to a specific firing position. This opening choice may have significant consequences, though, and feedback about them is required.

The goal in tactical training is to teach students to make good decisions in novel circumstances. To shape this behavior, the student must learn the consequences of each of his choices. Therefore, the tactical training environment must provide the trainee with full knowledge of the results of the decisions made during the training event. Students initially need more information than would be available in the real world. Feedback helps them develop a sound base of experience upon which to build. Proficiency comes about by trying something, seeing what happens, and remembering the results.

The tactician learns about results in actual combat as well, if the first encounter is survived. However, the amount and quality of information available upon which to base the next choice is often limited, and always less than desired. Learning to be tactically effective under these limitations is a critical skill, and its practice and development implies a learning environment with a significant level of cognitive fidelity, or fidelity of information.

High fidelity of information is equivalent to duplicating the "fog of battle" for the trainee. There is never really enough information, but what is available can be exploited at some level. The tactical training environment should be designed so that the information available at each step in the problem may be realistically reduced to levels that mimic field conditions.

DEFINING THE TACTICAL TRAINING SOLUTION

To this point, it has been established that the tactical training environment should provide a large set of novel problems, realistic information, a means of controlling the quantity and quality of information available during the problem, and a capability for fully examining the problem without information constraints at the conclusion of the exercise. This section discusses ways of engineering this training environment.

Clearly, getting tactical experience by any but synthetic means is too costly, in every way. The argument for simulation has always been the safety and relative cost savings with which critical experience could be gathered. The push for higher and higher fidelity is grounded in this argument, and in one other factor: we often have been unsure just how we are helped by experience; we only know that we are. The solution of a tactical problem involves selecting and implementing a series of specific actions. The environment must allow for this to take place. Therefore, an interface must be developed which allows the student the range of actions that are available in the operational environment. This can be achieved both in simple and complex training environments.

Many of the training requirements for tactical decision making can certainly be met in a full Weapons Systems Trainer (WST). For example, a pilot in a WST employing a air-to-surface missile may be given a surface target on his radar display and direction from his controller. He is briefed that, of the two significant radar targets he sees, the nearer one is a support ship. First the student sets up the weapon system for missile employment. He uses a "way-point" navigational option to position the missile flight path so that when the missile seeker activates, it will point at the combatant and avoid the support ship. During this evolution he is flying the aircraft, navigating, communicating with his controller and/or other aircraft, monitoring aircraft systems status, etc. After the missile is launched, the WST weapons scoring features might indicate that the missile hit a nearby oil platform instead of the intended target. Did the pilot have any other options? Could he have turned the missile seeker on earlier/later? Would a different navigational approach have worked better? Did the wind cause major cross track errors that he might have compensated for by using an offset targeting point? Did the

rain cause a delay in seeker acquisition? The trick is to understand which of the above additional complexities caused the miss. How does the pilot now learn to correct his mistake?

The WST will evaluate aircraft and missile parameters and determine whether the missile impacted the ship. Everything will function as it would in the actual aircraft on an actual mission, but although full fidelity of action was maintained, most of the training time was spent on performing the procedure to launch the missile. This in itself is certainly valuable. WSTs and similar devices allow tactical task performance under conditions of full mission parallel task loading. They cap the pyramid of the systems and procedures training hierarchy. At that point in the training system, they are the right tool for the right job. But they are probably not the best environment in which to begin the practice of tactics.

The complexities of making the correct decision in the employment of a sophisticated missile are not really addressed in the environment of a full system simulator. The pilot cannot see the effect of each of the parameters. As more are added, the situation becomes more complex.

The characteristics required of an effective tactical training environment follow directly from the training requirements themselves. Table 1 presents the implied training capability needed to meet the performance requirements of each of the 12 tasks listed earlier.

A PHASED APPROACH TO TACTICAL TRAINING

Phase 1 - Basic Decision Making:

What the table shows is that the decision making task is one of selecting and interpreting information. The actions taken as a result of this information must be selected and their results monitored. It does not follow that in order to begin learning decision making, a student must simultaneously perform weapons employment and platform handling skills. The student must know weapons and platform capabilities, and how they can offer up additional options, but for training the decision-making task, there is no further requirement other than to select these actions as a part of a tactical solution.

From this set of functional requirements, it is clear that minimum capability can be provided with a computer-based training (CBT) environment. One of the common objections to this approach is that typically the interface consists of multiple choice questions. While this might serve for very basic skill development, the small number of fixed choices is seen as making the problem too easy to solve; as we've noted, the real world set of options is open-ended, and imposing order on this chaos is a key skill to be developed.

The CBT environment can be made more suitable simply by increasing the number of options available. While a multiple choice question with only a few (less than 10) options requires the student only to recognize the correct answer, a really

Table 1.
Training environment requirements
for tactics decision making tasks

1.	Gather and analyze problem information	Sensor, intelligence information, platform status, weapons status, etc.
2.	Generate alternate problem definitions	As above
3.	Define the problem	As above
4.	Identify classes of appropriate actions	Reference material, capability to register selection
5.	Define areas of uncertainty	Sensor, intelligence information, platform status, weapons status, etc.
6.	Generate possible alternative solutions	Capability to register selection
7.	Search for/gather information to evaluate solutions	Sensor, intelligence information, platform status, weapons status, etc.
8.	Project solution consequences	As above
9.	Select a tentative solution	Capability to register selection
10.	Implement first action toward solution	Control of platform, weapons
11.	Evaluate results	Sensor, intelligence information
12.	Select follow-on action	Control of platform, weapons

large number of options, perhaps 20 or 30, requires the student to recall principles, select and apply rules, and use other strategies to reduce the number of options to a manageable level.

Phase 2 - Decision Making in Real Time

Moving on from fixed-choice CBT, a modeled simulation within a 2-dimensional framework permits greater fidelity of information by adding a real time clock to the problem. The criteria for performance of each of the tasks listed is clearly time-related, and with a different pace depending upon the platform. For example, selection of modes and parameters for launch of a missile from an aircraft performed as the range is closed must be done much more rapidly than if the same distance were being covered by a ship. Regardless, the time factor is real, and can be added through a "free-play" model implemented through a two dimensional interface.

Phase 3 - Adding Key Psychomotor Tasks

Added to the time pressure criterion are the constraints imposed by the conditions of performance. While the WST represents the full conditions of task performance, some of these conditions can often be cost-effectively implemented in part-task training devices. Airlines have used flight management systems trainers in this role for some time, and similar devices have been fielded for tactical training on specific weapons. These devices add some specific psychomotor conditions of performance at critical points in the decision making process. For example, slewing an electro-optical seeker head during the last stages of an attack in order to impact a specific vulnerable point is certainly the result of a tactical decision, but is made in real time from visual data realistically displayed, and using a slew control with a certain lag time and resolution limit.

Phase 4: Full job performance

Finally, the full job-oriented set of conditions and standards for tactical decision making tasks must be applied. This is where the full weapons system trainer comes into its own. By this time, the decision making process will have been practiced in relative isolation from parallel psychomotor task requirements - just as psychomotor skills will have been practiced without the pressure of decision-making - and the readiness for integrated performance in both domains will have been established.

SUMMARY AND RECOMMENDATIONS

This paper has presented a view of tactical decision making training that points toward a building-block approach to tactical skill development. This approach follows a process similar to psychomotor and procedural skill development, but is based on the information-processing and associated cognitive fidelity requirements of tactical decision-making. A suggested sequence of training events for this skill was described:

1. Open-ended decision-making practice in a frame-based (CBT) environment.

2. Real-time problem solving using phased levels of information fidelity in a two-dimensional setting

3. Addition of psychomotor and procedural performance requirements that are critical to tactical performance in a phased and controlled manner

4. Practice in full job performance in a high fidelity, high parallel task loading environment

The approach offers promise in developing tactical training both for specific weapons and for the tactical employment of full weapons systems.

REFERENCES

Van Gundy, A.B.: Techniques of Structured Problem Solving
New York, Van Nostrand Reinhold, 1981.

ABOUT THE AUTHORS

Jerome S. Bresee is a training analyst and designer with over 15 years of experience in training requirements analysis, training system design, and training capability development. He is employed by Delex Systems, Inc., where he is presently contributing to a large scale tactical training development program.

Michael E. Naber is a retired naval officer and A-6 Pilot initially hired by Delex Systems, Inc. to support weapon and aircraft integration programs. Mr. Naber is currently the Training Division Manager for Delex, responsible for multiple programs, most of which support the U.S. Navy and international customers by providing computer based tactical decision making trainers.

INTEGRATED TRAINING AND REUSABLE SIMULATIONS

David G. Fish
Robert H. Harter

Loral Aerospace Corporation
Hanover, Maryland

ABSTRACT

Although electronic simulations are effective training tools for many different applications, they often exhibit two significant limitations: 1) they do not provide an integrated training environment, and 2) they are based on a concept that is not reusable for the implementation of additional systems. During the development of courses used to train operators in the use of prime equipment, we developed an approach to overcome these limitations. The approach involved a simulation methodology that would be reusable for different systems, in which each simulation would encompass features similar to those found in commercial computer-based training software.

A team of software engineers and instructional designers developed an environment in which students progress naturally from novice to advanced operator. This progression is accomplished by providing the beginning student with many instructional prompts, helps, and supports, moving the student through incremental stages of less help and more independence; and ending the course with exercises that realistically simulate the prime equipment. These varying levels of instructional exercises are made possible by the use of table-driven data structures within the simulation software. This technique allows instructional designers to create countless versions of a simulation exercise.

This paper describes the common limitations of training simulations, the methodology we used to overcome those limitations, and the table-driven data structures at the heart of the simulators we built for a DoD program called GUESTMASTER. Additionally, we describe refinements we made after the initial implementation of this concept.

PROBLEM

Simulation-based training has a reputation for being expensive to design and develop. It is also criticized for a lack of built-in instructional controls. The problem we faced was to devise a simulator design that would be easy to maintain, applicable to multiple systems, and provide useful instructional controls -- including scenario selection, prompts and helps, and feedback based on trainee performance. We addressed this problem in one aspect of the GUESTMASTER program.

HISTORICAL PERSPECTIVE

The GUESTMASTER program encompasses the design and production of a 1,200-hour curriculum to train the operators of sophisticated electronics equipment. Two primary considerations determined the initial direction: because of mission complexity, students needed theory lessons to understand the operational context; because of mission sensitivity, students needed hands-on lessons to ensure accurate performance without extensive on-the-job training. Based on these two needs, and the expected volume of student throughput, our user directed us to design comprehensive courses using both computer-based training (CBT) and high-fidelity equipment simulations.

We allocated two types of training to the CBT portions of each course. pre-requisite skills and knowledge necessary for job performance, and tasks selected for training that were to be taught only to a knowledge level. We designated the remaining, procedural objectives of each course to be trained on high-fidelity equipment simulations (see Figure 1).

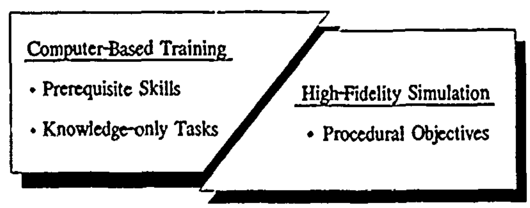


Figure 1. Allocation of Training Requirements

Use of CBT

From its inception, the GUESTMASTER program has delivered courses on the MicroTICCIT CBT system. This commercial CBT system provides the management and student evaluation data required by our user to oversee and monitor the effectiveness of the courses. This information has proved useful. As a result, the user mandates employing this system for all courseware. It has worked well in providing course exercise hierarchy management, instructional control of the simulators, and the provision of instructional feedback, therefore, it has not required modification from its original intent. For this reason, we do not address in depth the role of CBT in this paper.

Simulation Constraints

Because of the initial schedule and budget, our user directed us to proceed under two constraints. First, we could use only a specified hardware suite consisting of 1) a commercial CBT terminal; 2) the operator console portion of the prime mission equipment; and 3) an IBM PC/AT to provide an interface between the CBT and the operator console (see Figure 2). Second, we could use only operational software in the prime equipment, to reduce the cost of simulations and ensure that the final product would be identical with the fielded system in look, feel, and timing. This precluded the display of instructional messages on the operator console.

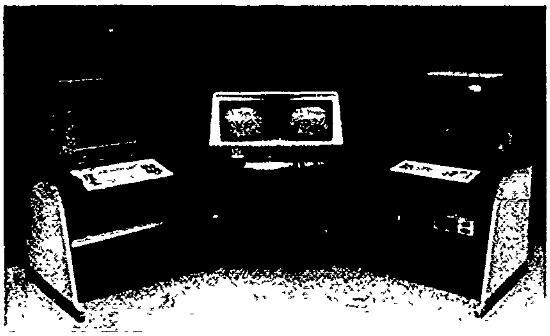


Figure 2. Training System Configuration

Early Simulation Projects

The first simulator that we deployed was built around operationally "smart" software; that is, most processing occurred within the console software (see Figure 3). Therefore, the console software had little need for external communication, which meant that the CBI courseware was unable to evaluate the correctness of student performance. The primary data reported outside the operator's console were updated equipment settings, so evaluation was based on trying to "guess" how the values were changed by examining these updated equipment settings.

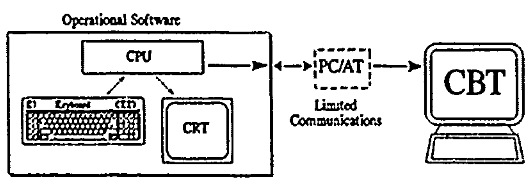


Figure 3. "Smart" System Architecture

The second simulator that we deployed, on the other hand, was built around operationally "dumb" software; that is, it required a precisely defined interface with a control computer. This fundamental difference meant that little of the first simulator's software was useful on the second simulator; the development team, therefore, was required to devise another approach. We simulated the responses of the control computer with software running on an IBM PC/AT (see Figure 4). This enabled the software to access some of the student's actions, which provided an improved, though still limited, evaluation capability. Student responses were first processed by the console software and then sent via interface messages to the simulated control computer. At that time it was too late to stop incorrect values from being processed because the internal subsystem variables were already updated beyond our control. Student errors required that the courseware terminate the exercise, provide appropriate feedback, and restart the simulator, thereby eliminating the instructional benefit of allowing the student to perform the correct action within the context of the exercise.

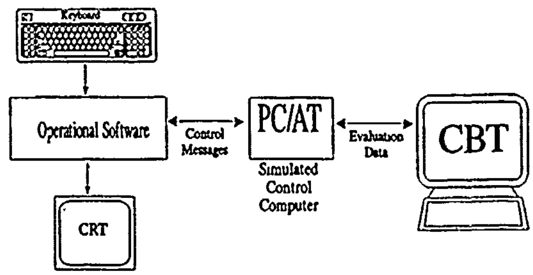


Figure 4. "Dumb" System Architecture

Maintaining Training Materials

While we were developing and testing exercises for both of the first two simulators, we discovered a tracking deficiency. We found that tracking minor content changes consumed far too much time because the required technical content was stored in four separate places (see Figure 5). Data that the student needed during completion of a procedure were stored in a student guide; content-related information and exercise directions were portrayed in storyboard format; interface commands were embedded in source code entered by a CBT programmer; and the machine instructions necessary to establish the initial content environment were stored in additional data structures.

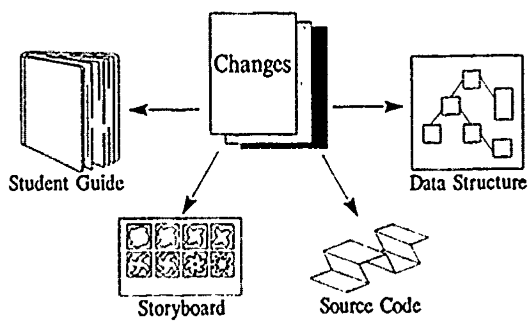


Figure 5. Locations for Training Material

LIMITATIONS OF EARLY COURSES

The difficulties we faced during the early projects hit us on several fronts: minor changes to completed exercises were cumbersome, software inheritance was lacking, meaningful student evaluation was scarce, completing an exercise without interruption was infrequent, and helpful messages on the operator console were prohibited. Consideration of these difficulties led us to the observation that they are symptoms of two larger limitations.

First, despite the accuracy and fidelity of the simulations, they required frequent courseware intervention to provide the student with the information needed to progress through the course. That is, they provided no "integrated training capability."

Second, as we prepared to add simulations of additional subsystems to our course curriculum, we found that the prohibitive time and expense necessary to create these new simulations may have led to their elimination altogether, eroding the effectiveness of the instruction. In short, the methodology of simulation development was not reusable.

These early experiences illustrated both the strengths of accurate and realistic simulation-based training, as well as the limitations inherent in the original constraints. We had satisfied the user's expectations, yet knew that further refinements of the training capabilities were possible. A new approach was needed.

A SOLUTION TO THE LIMITATIONS

We came to this conclusion at a critical point in the overall contract schedule. A total of five simulation-based courses had been scheduled for delivery; at this point in the schedule, two had been delivered, one was scheduled to begin production immediately, and two others were scheduled to begin production within a year. (see Figure 6. The highlighted area represents the scope of this paper.)

With our experience, we knew that we could not use operational software as the foundation for an integrated training simulator. We also knew that production schedules for the development of simulator software had to be trimmed. We decided, therefore, to chart a new path by programming the entire training system ourselves, to include both the simulation software and the integrated instructional capabilities. The time was right to try this new approach. although the instructional concept was unrefined and could require several iterations, the next deliverable course was on a small, comparatively unsophisticated system that had a lower expected student throughput. Given these factors and a few accompanying capability demonstrations, the user recognized the advantage of revoking the original constraint limiting us to using only operational software, as well as the benefits of using this small third course as a test bed for training and software personnel to concentrate their efforts toward the development of a long-range solution. The results of the "test"--successful development and deployment of this third course and simulator--proved that we were on the right track.

We then began production of the two larger courses. As we worked toward overcoming our two limitations--little software reusability and no integrated training--we realized that separate resolutions would not be necessary. It became apparent that taken and applied individually, each could improve a course; fitted tightly together, they result in an overall design that allows the development of full-scale simulation trainers. We describe these two concepts in the following sections.

SIMULATOR	SCHEDULED START OF PRODUCTION	OPERATIONAL SOFTWARE	EVALUATION CAPABILITY	COURSE SIZE
First	Delivered	"Smart"	Very Low	Medium
Second	Delivered	"Dumb"	Low	Very Large
Third	Immediately	Replace	Medium	Small
Fourth	One Year	Replace	High	Large
Fifth	One Year	Replace	High	Large

Figure 6. The Critical Point in the Program Schedule

THE CONCEPT OF REUSABLE SIMULATION

Aside from our experiences, we had no guidance as we ventured into uncharted territory. But this much was clear: whatever concept we devised would have to be reusable for multiple systems that were to be simulated during the remainder of the contract. Our concept of reusable simulation consists of the following: identify all possible student actions, list these actions within data structures, and combine these lists in a variety of ways to create simulation exercises. Because this fundamental concept can be reused across multiple courses, the development tools and major portions of the evaluation software can also be reused.

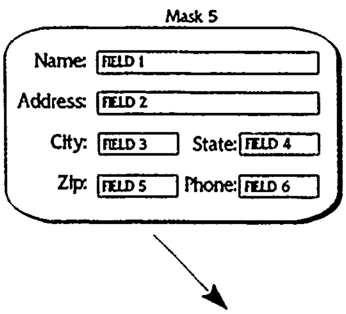
Identification of Student Actions

The first step required subject-matter experts to identify each procedural action that students would be expected to perform during the course. We assigned to each action a step identification (SID) number so it could easily be referenced by the subject matter experts and writers, as well as by the simulation software (see Figure 7). The types of steps include pressing keys on the operator's console, completing system prompts or displays, accessing instructional options, adjusting simulated peripheral equipment, and such system-initiated actions as generating error messages or automatically activating an equipment condition.

SID	Description
113	Access the SAVE MASK function
114	Clear error messages
115	Save configuration files
16	Abort operation and exit w/o saving
"	"
"	"
"	"

Figure 7. Step Identification List

In addition, each field or value that a student could enter or change was identified for software manipulation and evaluation purposes (see Figure 8). As a result, we assigned parameter identification numbers to each possible field the student could access. Once all steps and parameters were identified and named, the definitions of simulation exercises simply required establishing the appropriate relationships among the data elements.



Mask	Field	Description
5	1	Name
5	2	Street Address
5	3	City of Residence
"	"	"
"	"	"
"	"	"

Figure 8. Assignment of Identification Numbers

Data Structure-Based

Data relationships were defined by two primary data structures: the procedure table (PT) and the instructional record (IR).

A procedure table is a sequential list of procedural step identifiers (SID's) that the student is expected to perform (see Figure 9). For each step in the procedure table, any system parameters that will be evaluated are listed without a specific data value. An instructional record is a compilation of procedure tables and the associated correct answers for an instructional scenario (see Figure 10). In this way, procedure tables may be reused in different IR's without recreating the procedure definition (see Figure 11).

Procedure Table 37

Step	SID	Description
1	113	Access the save function
2	200	Complete prompt-field 6
3	237	Display error to student
4	114	Clear error messages

Figure 9. Sample Procedure Table

Instructional Record 213

Time Limit 16

First Procedure 37
CORRECT ANSWER 5

Second Procedure 56
CORRECT ANSWER 12

Figure 10. Sample Instructional Record

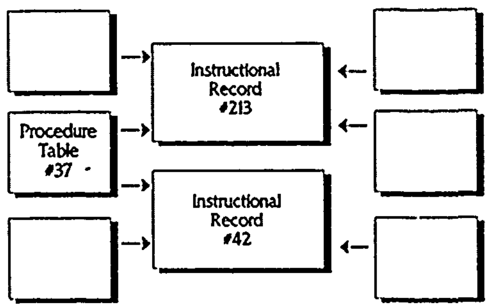


Figure 11. Reuse of Procedure Tables

Because of the complexity of the equipment we were simulating, large amounts of technical data were required for each exercise to function accurately. Approximately 20 supporting data structures were identified to provide maximum flexibility for the composition of instructional exercises (see Figure 12). In some cases, several levels of relationships were required to store and manipulate the required data. Some of the additional tables included data for several types of instructional messages, lists of user or system error conditions, instructional scenario text, correct data values, and common-error relationships.

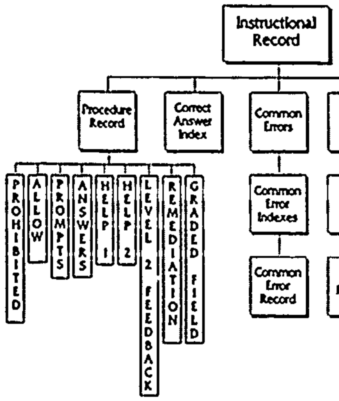


Figure 12. Data Structure Relationships

In the development of the first two simulators, the entry and tracking of this volume of data was time-consuming. Our new approach simplified the maintenance of training materials, making production more manageable and cost-effective.

Creation of Simulation Exercises

The elaborate interrelationships described above allow the creation of simulation exercises. To provide efficient organization of the data structures and to minimize the data entry time, we built an "IR/PT Editor" around a commercial database management system (DBMS) to allow users to define the interrelationships. This editor includes capabilities for multiple types of reports, in addition to producing ASCII files that can be transported to the LSI computer.

The report capabilities encompassed in this commercial DBMS also allowed us to eliminate data re-entry. Formerly, data was entered and stored in four different formats and systems. Now, it is stored in a single location to facilitate the modification of course content (see Figure 13).

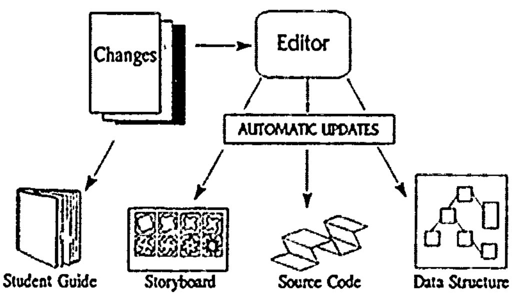


Figure 13. Upgraded Locations for Training Material

Mode-driven Operation

Finally, to make the most cost and instructionally effective use of the data structures, we defined different simulator "modes" of operation. The differences between the modes of operation center on issues such as the number of attempts given to the student to correctly complete a procedural step, the availability and level of detail present within the instructional messages, and the availability of optional features such as suspending the current exercise.

This use of operational modes allows the student to progress from a directed performance of each procedural step, through two intermediate stages, to a nearly free-flowing exercise. This progression is achieved through use of four training-defined, software-implemented modes (see Figure 14, which shows one mode).

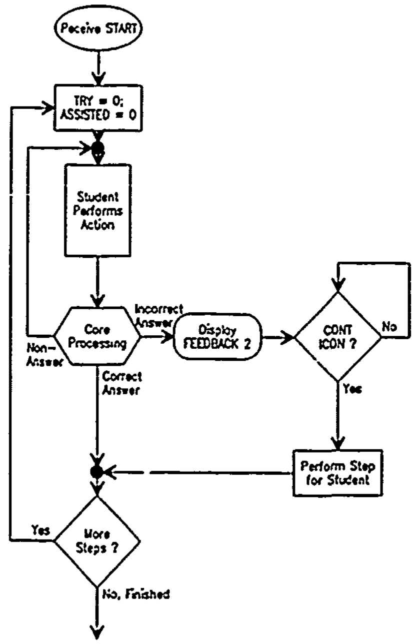


Figure 14. Example of Operational Mode Logic: Test Mode

An additional mode provides the courseware with the capability to place the student in a specific environment, with all displayed and underlying fields set to their appropriate values, and with the proper presentation on all displays. For example, this feature can be used to provide remediation on a particular step with which the student may have had difficulty--with minimal time delay and pinpoint accuracy.

The most important idea behind the modes of operation is that the definition of each mode may be changed based on the instructional requirements of a given course. A situation where performance of procedures is

not critical or life-threatening requires less-stringent mode definitions than a case where the accurate performance of a procedure is determined to be life-critical.

THE CONCEPT OF INTEGRATED TRAINING

The need to develop the concept of reusable simulation symbiotically fed into the need to improve the instructional effectiveness of our courses. In order to provide the reusable framework, we programmed the prime equipment simulation ourselves; this in turn allowed us to introduce training features directly into what otherwise appears to be operational software. These on-line training features--instructional messages and student-selectable exercise control options--are at the heart of what evolved into the concept of integrated training. To incorporate this concept into a course, two aspects must be clearly defined: the equipment to be simulated, and how the desired training features will be integrated.

Simulated Equipment

Since all of our large simulations used the same console, the primary hardware platform was already established. However, each job involved additional equipment that required training. Our decision was to use an IBM PC/AT and a 19-inch touchscreen monitor to simulate this peripheral equipment (see Figure 15). To reduce development costs, we decided to simulate only those functions that were going to be trained, rather than simulating all operational subsystem functionality. This decision applied to the peripheral equipment simulated on the 19-inch monitor, as well as the simulation of the operations console.

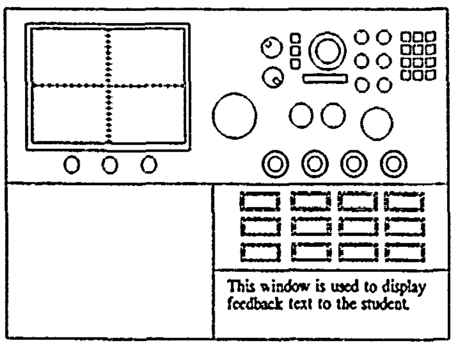


Figure 15. Peripheral Equipment Simulation

However, we found that the 19-inch monitor's resolution was not high enough to display both the complex peripheral equipment images and the desired on-screen instructional features. As a result, we developed a zoom capability in which the student is allowed to touch any portion of a simulated equipment faceplate and have the corresponding area enlarged in a "zoom window" on the bottom of the screen (see Figure 16). It is in this zoom window that the student can manipulate the functional controls. Updated settings are dynamically displayed within the zoom window and on the simulated CRT as changes are made. In addition, updates showing the resulting knob positions are displayed on the faceplate graphic when the student zooms another window.

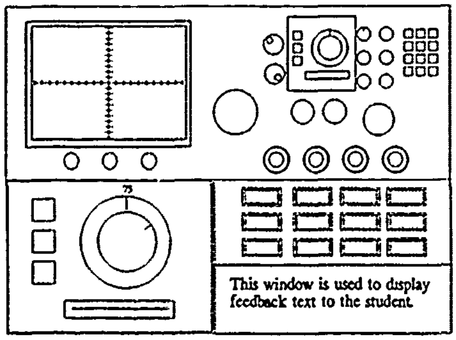


Figure 16. Use of Zoom Zones

Instructional Features

Textual Messages. All subsystem-specific messages are accurately simulated in both their content and their reason for appearing. We supplemented this set of subsystem messages by adding the ability to display immediate feedback for incorrect responses or other errors. The messages appear on the simulator and do not require the student to return to the computer-based instruction. The textual messages we display are of two types: directive and instructional.

Directive messages provide the student with information about the instructional environment. They inform the student about what is happening or tell the student what to do next. Instructional messages provide text relevant to the current scenario or exercise. This can be in the form of help, feedback, or remediation; prompts for the next action; or statements containing the correct answer.

An important part of the concept of integrated training is that all messages are displayed on the screens where the simulation takes place. To accomplish this, we defined several areas on each operator console CRT as pop-up windows. Each window is displayed with a highlighted border to indicate that the contents have been added for instructional purposes and are not part of the operational software (see Figure 17). The specific window to be used for a given step is selected by the instructional designer or subject matter expert when the exercise is created.

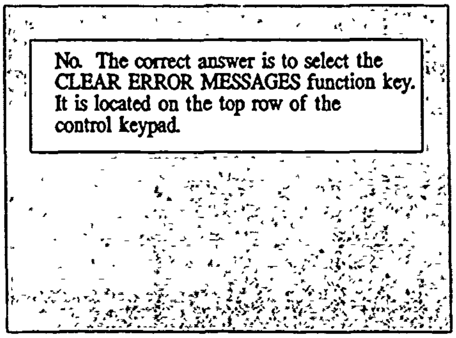


Figure 17. Instructional Message Highlighting Scheme

Although the display of textual messages on the subsystem screens eliminates the need for the student user to constantly return to the CBT terminal, it causes the obstruction of subsystem displays. To remedy this situation, we provide a capability for the student to cycle messages among the different windows (see Figure 18).

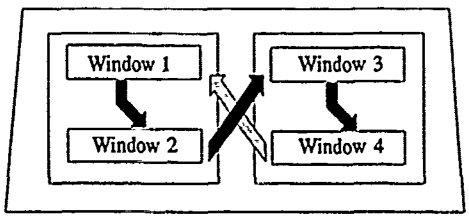


Figure 18. Cycling of Instructional Messages

Instructional Options. We defined an instructional options menu (IOM) to provide access to student-selectable exercise options. The options include accessing various levels and types of help, feedback and remediation messages, paging through instructional message text; accessing equipment simulations, suspending the current exercise; returning to the CBT workstation for additional information; and initiating free-form evaluation of graded parameters.

On the first integrated simulation trainer we built, and the third simulator overall, only subsystem console functionality was provided. We later realized that teaching both procedures and decision-making skills often requires the operator to perform actions away from the console or to enter data on an additional workstation, for example, using a telephone to call a superior, or checking a smoke detector or circuit breaker located on the other side of the room. The student can select from a textual list of "non-console" functions and is evaluated based on the correctness of that choice.

Evaluation Flexibility. An important aspect of a training device is its ability to match the procedural sequence requirements of the trained system. That is, when the trained system requires a strict step-by-step sequence, the training device must require the same order; when the trained system permits flexibility, the training device must permit flexibility. This order variability must be supplied while evaluation control is maintained. To provide this combination, we developed the idea of "order factors." Order factors, which are an integral part of the IRs, provide the course designer with several options: a procedure can be required at a specific point within an exercise; a procedure can be allowed to be performed at any time within an exercise; or a set of procedures can be linked such that only one of the procedures may be performed. When an instructional designer or subject matter expert combines these factors appropriately, we are able to create training exercises that simulate how the trained system operates and at the same time maintain strict evaluation control.

Prohibited and Allowed Lists. To accommodate variations in the sequence of procedural steps, we designed the data structures to contain lists of steps that are either prohibited or allowed. If the student attempts to perform a prohibited step, a message is displayed to indicate that the requested action may not be performed at this point in the current exercise. Student actions that are on the allowed list are executed by the simulation software as in the operational environment. An attempted action that is on the allowed or prohibited list does not affect the evaluation of student performance. The contents of these lists are under control of the instructional designers for each step in each procedure.

Evaluate Icon. Together with the allowed step list, we use an additional technique to deal with procedural variations. We added an option that will postpone evaluation until the student selects a particular option. In this way, a "step" is defined as a discrete action, a limited action, or a free-form action. After performance of the action(s), the student requests evaluation and is graded. This accommodates situations in which minor adjustments must be made to the equipment until the desired result is obtained, such as fine-tuning a receiver.

Common-Error Analysis. Instruction is only as good as its ability to deal with errors. Often, specific errors are common among students. These common errors must be explained and remediated in a meaningful way. To accommodate this capability, we added data structures to allow a comparison between each student action and the actions that technical experts determined may be common errors. If this comparison results in a match, alternate, more explicit, instruction is provided.

Evaluation Algorithm. Another critical instructional feature is the algorithm used to evaluate a student action. Given that a student action can be classified as one of several possible types, we determined a meaningful sequence of evaluation based on the types. (see Figure 19).

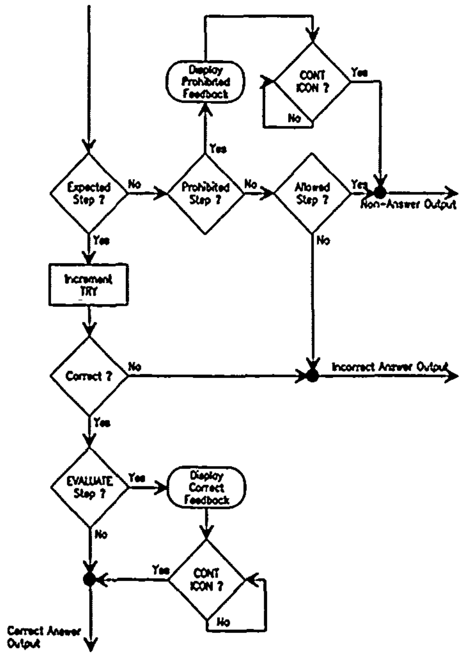


Figure 19. Core Evaluation Processing

First, we examine each student action to determine whether it is valid from a subsystem viewpoint. If it is not valid, we simply provide the same feedback as the operational software and allow the student to try again without penalty. This is essentially what happens in the real environment and therefore is instructionally sound given our training requirements.

Second, if an action is determined to be valid from a subsystem standpoint, we check to see whether it is the correct answer. In other words, a particular value may be acceptable in a given field, but not correct for the current scenario. If the response is correct, the student is then expected to perform the next step in the procedure table; if the response is incorrect, the software executes the next check in the evaluation algorithm to determine if another response type was entered. In all, the evaluation algorithm checks for six possible types of responses and deals with them accordingly. The last step is to score the student as incorrect and to take the appropriate actions for the current mode of operation.

REFINING THE CONCEPTS

When we tested the first course developed with these new concepts, our users liked what they saw. Many of the discrepancies noted in the first two simulators were eliminated: exercises were not interrupted following an error, tracking of content changes was more straightforward because of the data structures and the IR/PT editor, and feedback appeared on the operator's console at the student's point of contact. However, our user also identified one glaring weakness, with which the students readily agreed. Because of the complex scenario setup that often was required to place the student at the appropriate instructional location for a given exercise, the time required to perform the setup could be unbearably long -- an average of eight minutes. The user required that this time be reduced.

Modified Pre-set Concept

The development team determined that much better performance would be obtained if, instead of proceeding through the set-up data structures step-by-step, all of the simulation's internal values are read in from a "pre-set" file. A set of utilities was developed that allows instructional designers or subject matter experts to perform procedural steps and write the final state to a pre-set file. During exercise delivery, the CBT system can request that any pre-set file be loaded. The CBT system then specifies the corresponding instructional record to use in tracking the student's progress on the simulator. The start-up time is now close to constant, since the information needed to store the state of the system is nearly the same from one scenario to the next. Instead of the original eight minutes, all presets now occur in less than two minutes.

Completing the Cycle

The pre-set capability was in place for the subsequent three-week pilot test. Our user was more than satisfied with start-up performance, and with other minor improvements. The course was deployed, tested, and delivered to the schoolhouse site, the same site where our early courses are used by dozens of students every month. The instructors were impressed with the performance of the newest course, both with the start-up time and its instructional effectiveness. Based on this success, the user has asked that we convert the most-used simulator and course (the one that uses "dumb" operational software) to this integrated training architecture.

This "conversion" work has given us another opportunity to further develop the concepts described in this paper. Specifically, work is already underway to streamline the debugging of IR/PT data by developing an on-line, networked test facility. This further work illustrates that although the concepts of integrated training and reusable simulation have been effective in providing high-quality training tools, we look forward to the benefits of further refinements.

CONCLUSION

Internal instructional control of student progression and meaningful student evaluation are long-sought goals of simulation-based training. The training value of simulation exercises is significantly enhanced by the incorporation of such capabilities. In effect, the GUESTMASTER program has shown how principles of simulation can be married with CBT-style control to yield simulators that teach. More importantly, the basic design is reusable across simulations, making each subsystem easy to maintain and simplifying the task of adapting the model to other requirements. Careful coordination between instructional designers and software engineers yields a design that reflects the best of both worlds: simulation that performs with high fidelity while providing guidance within a strict instructional framework.

As systems grow increasingly complex, questions about the relationships between effective instruction and simulation will grow proportionally. We believe that the solution described in this paper provides some basic guidelines for answering these questions.

ACKNOWLEDGEMENTS

The authors would like to thank Dr. Frank Capuzzi for his help and encouragement during the development of both the original concept and this paper; Joseph Mintzer for his technical assistance in the preparation of this paper; and Randolph Terlecki for his graphics expertise.

ABOUT THE AUTHORS

David G. Fish is the Senior Technical Expert in Simulation and Programming for the Instructional Systems Department of Loral Aerospace Corporation. He has worked on the simulation concept and implementation of five full-fidelity simulators. He has also served as a lead programmer for the production of computer based training. Prior to this assignment, he was the East Coast photographer for Loral and previously worked on developing multi-media training resources. He holds a B.A. in photography from Shepherd College, and is currently working on a masters degree in computer science at Johns Hopkins University, Applied Physics Laboratory.

Robert H. Harter is a Software Engineering Specialist in the Systems and Simulation Section of the Software Engineering Department at Loral Aerospace Corporation. He has been involved in the production of training-oriented software since 1983. He has participated in the design and development of five full-fidelity software simulations, and has worked closely with the instructional staff to improve the effectiveness of software within the integrated training environment. He holds a B.S. in Computer Science and Mathematics from Vanderbilt University.

APPROACHES TO AIR TRAFFIC CONTROL/AIR DEFENSE WORKSTATION SIMULATION AND TRAINING
(CATEGORY: TECHNICAL)

Walter Sobkiw

1700 Tanglewood Drive

St. Petersburg, FL 33702

ABSTRACT

A commonly desired approach to supporting air traffic control (ATC)/air defense (AD) embedded training and simulation activities is to provide the equivalent of a fielded system with additional hardware and software to support scenario generation and target generation functions. The problem with this approach is that the ATC/AD workstations are costly subsystems and, during the later stages of the systems life cycle, organizations attempt to meet increasing training needs with low cost solutions that are not equivalent to the fielded systems. Further, the training and simulation requirements of the organization tend to merge and simulators used for system upgrades and system studies are tasked with also providing training services.

This paper presents two contrasting approaches to providing generic ATC/AD training and simulation workstations. The first approach was implemented by the FAA at the FAA Technical Center in the early 1980's using workstations with removable bezels and shelves. This was a hardware intensive approach with custom software to duplicate the existing En Route and Terminal Radar Approach Control (TRACON) ATC consoles. The second approach is based on current technology using Reduced Instruction Set Computer (RISC) workstations, Rapid Prototyping MMI Software, and variable function keys (buttons) on the monitor(s) to simulate the "knobology" of target workstations. Both systems are contrasted from cost, complexity, and operational efficiency points of view.

BACKGROUND

The traditional ATC/AD workstation has been a combination of mechanical packaging and computer controlled display and entry devices (figure 1). The mechanical packaging has included overhead backlit maps of airspace, devices to house paper flight strips, and provisions for supporting voice communications panels. The computer controlled display and entry devices have provided a picture of the air situation, which includes tracks, various maps (geographic, air routes, sectors, etc.), tabular displays, and control of the air situation display. The control of the air situation display has included display filters for information "decluttering" and system input such as track handoff. Each system has varied in the characteristics of these two basic elements: mechanical packaging and computer control.

The current FAA En Route and TRACON centers are good examples of how the ATC workstations vary. The primary mission of each system is the same, to provide safe and efficient air traffic control services. However, as the system evolved, different

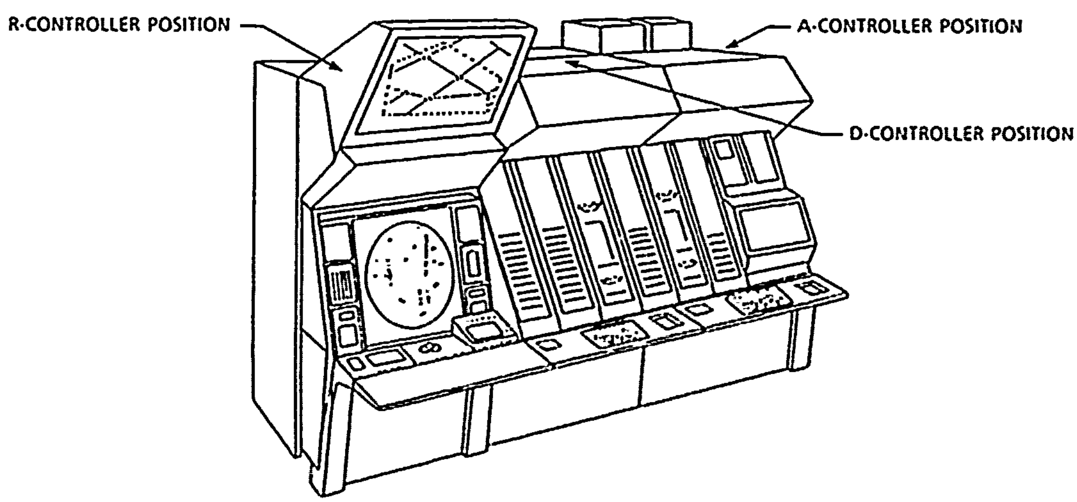


Figure 1. En Route ATC Sector Suite Workstation

technologies were applied to the En Route and TRACON workstations. With the current Advanced Automation System (AAS) program the goal is to provide the same basic workstation for En Route and TRACON operations.

In the current US ATC system, air traffic controller training has taken these system characteristics into account. An air traffic controller's training begins with classroom instruction, moves to hands-on simulation training at the FAA's national training facility in Oklahoma City, and continues with hands-on training at an operational site using site simulation/training capabilities and on-the-job exposure. The first time an air traffic controller "sees and feels" the actual operational equipment is at the site. This situation is further complicated in that the FAA Technical Center may be called upon to support training such as during the controllers' strike during the early 1980's.

The FAA Technical Center is responsible for test, evaluation, and support of all systems, including the En Route and TRACON facilities. To support the En Route and TRACON facilities, the FAA Technical Center houses two groups of laboratories. The first group of laboratories is the National System Support Facility (NSSF) formerly the Air Traffic Control Simulation Facility (ATCSF) and it is responsible for "Test and Evaluation" of "Advanced ATC Concepts." The second group of laboratory facilities is the En Route and TRACON support facilities which are responsible for maintenance of the fielded systems. These labs prioritize study, training, and field maintenance activities based on their primary missions.

EMULATION AND TRAINING CONSOLE APPROACH

Hardware Emulation Approach - Hardware Defined Removable Bezels and Shelves

In the late 1970's the FAA was looking to upgrade the ATCSF. All these background issues and system characteristics were present and impacted the upgrade approach. The old ATCSF consisted of displays that did not match either En Route or TRACON workstations. One of the goals of the upgrade was to provide workstations in the ATCSF that "looked and felt" like En Route and TRACON displays. It was believed that the ATCSF could then produce more meaningful emulations for system human factor studies and "occasional" training activities.

To use the actual field equipment in the ATCSF was cost prohibitive. Technology had progressed since the introduction of the fielded En Route/TRACON displays and a lower cost alternative was available. The approach was to use the Sanders Graphic 7 stroke display processor driving a 20" round beam penetration (4 color) CRT and removable bezels and shelves that would interface to the Graphic 7 internal PDP 11/04. These bezels and shelves were manufactured to look and feel like En Route and TRACON bezels and shelves (figure 2). They used the actual parts from the En Route and TRACON radar display position. The panel switches fed a custom switch scanner and the "Pots" (variable resistor controls) fed a custom Pot scanner. The trackballs were even mechanically modified to provide the feel of either an En Route or TRACON trackball. The En Route trackball sat on a bearing assembly and "rolled" with the swing of the hand while the TRACON trackball sat on a pedestal and stopped instantly without hand movement.

The surface area for the TRACON trackball was even reduced with the use of a cover plate to match the surface area of the fielded trackballs. Attaching an En Route bezel and shelf would activate En Route software in the console and the console would then emulate the En Route radar position. Attaching a TRACON bezel and shelf would activate TRACON software and the console would emulate a TRACON radar display. Attaching an R&D bezel and shelf would activate R&D software in the console and emulate the next generation ATC console.

There are several advantages to the generic console implemention using the removable bezels and shelves. The first is that the console's physical configuration can be emulated as opposed to simulated and the emulation can be accomplished with flexible software and low cost bezels and shelves. This would permit the operators in training to not only become proficient in the functions of the console but also imprint the physical characteristics of the console on the operators such as tactile feedback, reach, and field of view. This was an excellent solution for satisfying some of the goals of the ATCSF. Both En Route and TRACON radar position consoles were emulated and studies of new man machine interfaces (MMI) could be supported with the introduction of other bezels and shelves.

At the time some of us at the FAA believed we were developing a prototype for the technology to be used in the next generation ATC workstations. On-the-fly text generators replaced monoscopes. Computer control replaced mechanical controls such as brightness. Beam penetration CRT's would provide color. However, while supporting the Hughes DCP AAS program, we clearly saw that the next generation ATC console would be based on technologies with different foundations. High resolution 20" × 20" 2048 × 2048 raster color would replace stroke technology. Programmable variable and fixed CRT displayed function keys would replace switch panels and other mechanical devices.

SIMULATION AND TRAINING CONSOLE APPROACH

Software Simulation Approach - Software Defined

Figure 3 is an example of an ATC/AD workstation based on a commercial Reduced Instruction Set Computer (RISC) hardware and a software package that supports MMI implementation using rapid development/rapid prototyping tools. The switch panels of the previous generation ATC/AD workstation have been replaced with fixed and variable function keys displayed on the CRT. These keys could just as easily be implemented as picture representations of knobs and toggle switches. The mouse or trackball is used to control any of these CRT displayed switch panels, knobs, toggle switches,

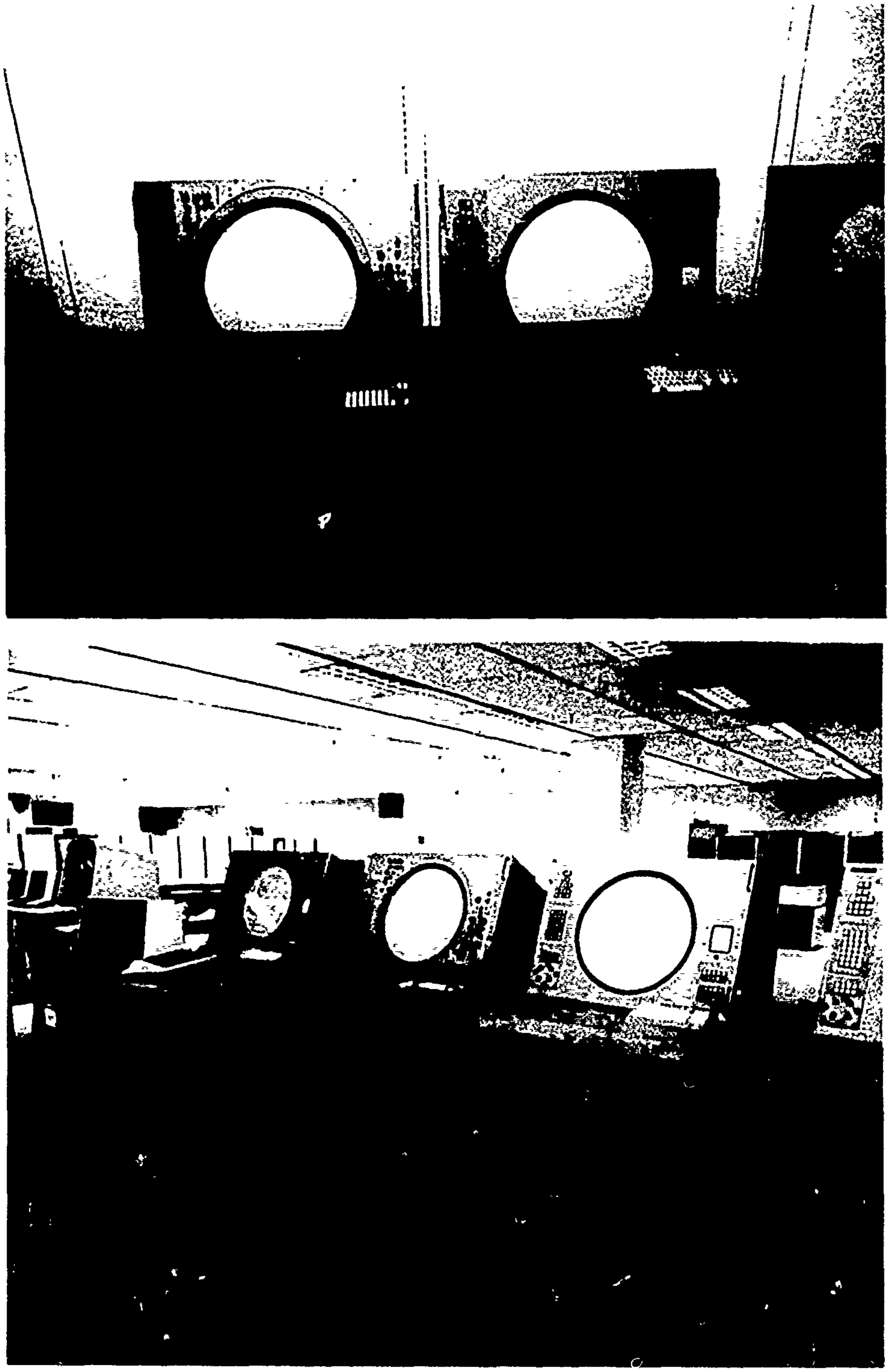


Figure 2. En Route and TRACON Bezels and Shelves

or other operator entry devices. The switch panels and windows on the situation display are always placed such that the situation display will not be covered with lower priority data. The main menu area calls up different switch panels such as Category Select and Feature Select. When a button is activated, it is highlighted and the appropriate function is performed. Tracks can be hooked with either the keyboard or the mouse.

The main menu bar is used to select the "virtual" panels, switches, and knobs found on the actual operational console. In this example, the main menu

supports 12 buttons that are approximately .75" square. A main menu bar button supports 3 rows of 5 characters each to define its function. The submenus contain 6 columns × 9 rows of buttons for a total of 54 buttons. The submenus measure 2.75" × 4.75". The intent was to use 56 pixels in the horizontal direction of the 1280 × 1024 resolution display to provide a situation area of approximately 1024 × 1024 pixels. The current colors of the menu areas are two shades of blue with black text. When a "button" is selected, it turns green to show activation. The following submenus are provided as an example of an Air Defense workstation:

Category Select - Filters static areas on the display and filters broad categories of track data.

Feature Select - Filters data blocks shown on the display and filters tracks based on speed and altitude.

Range Scale - Provides traditional range scales and nontraditional features which permit the selection of predefined centers and range scales based on console defaults and air base locations.

Track Control - Provides the ability to initiate tracks, assign tracks, drop tracks, and create manual tracks.

Pilot Control - Used in simulation and allows the simulation pilot operator to control up to 10 airplanes.

Console Control - Permits the operator to create special areas, print tabular or situation data, and set the local time of day.

DQM Control - Permits various radar data to be filtered from the Data Quality Monitor (DQM) display, and provides various system management functions such as setting the system time.

This type of trainer can be developed a number of different ways. If specifications on the original system are available, those specifications can be used as the foundation for describing the trainer's MMI. E-Systems developed three separate MMI's using a storyboard method. In this approach all the keyboard entries, menu's, submenu's, and display results are shown on a standalone piece of paper. Many of the functions are well understood. Little explanation was required behind the underlying function of a "button" on the display and the resulting display symbology and text. If the underlying functions are new or complex, the appropriate paragraph numbers can be referenced in the original system specifications. This was not required when E-Systems developed the three separate MMI's (over-the-horizon surveillance, missile warning, and long-range radar air space management).

The primary advantage to this training workstation approach is the fast turnaround time for being able to simulate any ATC/AD workstation. On each occasion, E-Systems has been able to implement complex "air" surveillance MMI's in less than two months. An internal target simulation capability can provide the scenario to support various levels of training. This, coupled with the relatively low cost commercial work-station hardware, can provide a desktop trainer that can support all the functionality of any ATC/AD work-station. The limitation to this

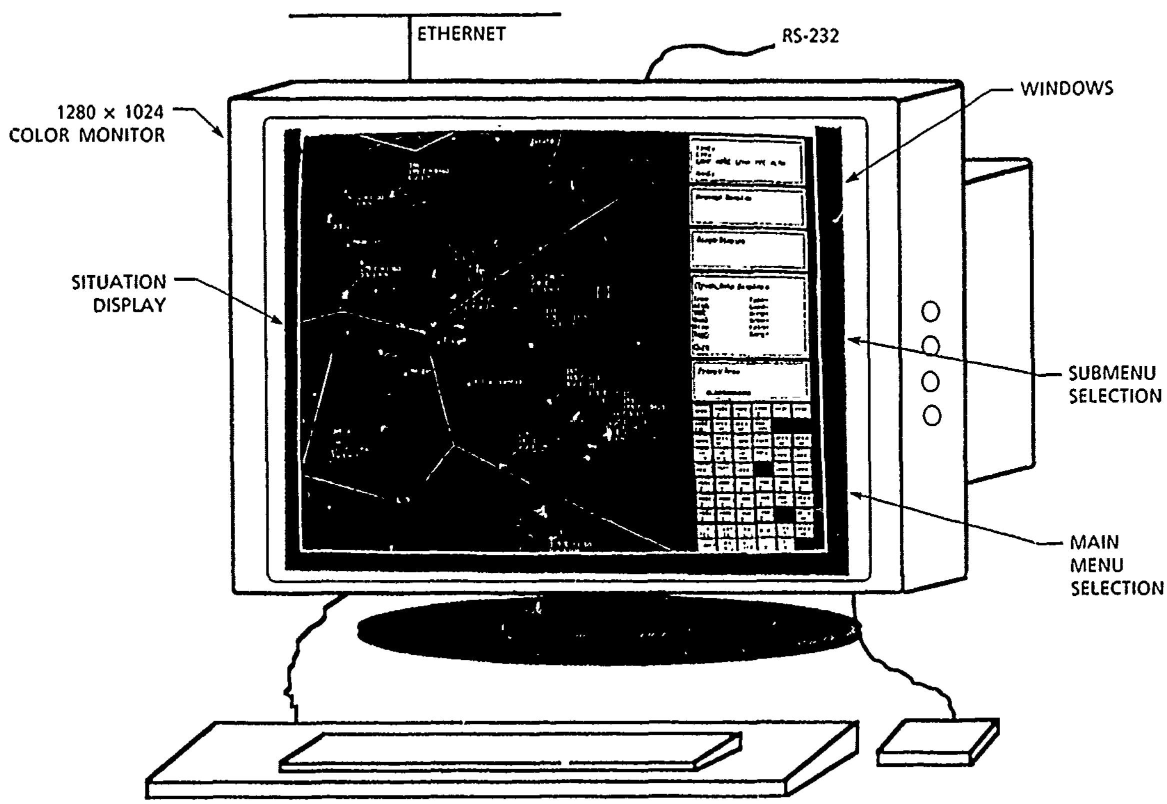


Figure 3. ATC/AD Workstation

approach is the ability to emulate the physical characteristics of the workstation such as the tactile feedback, reach, and field of view.

This CRT based trainer workstation taps into several current trends. The first trend is towards workstations based on fixed and variable CRT displayed function keys such as in the case of the Advanced Automation System (AAS) sector suite. The second trend is in the growing automation of commercial tasks. Many current commercial graphics applications requirements outstrip traditional ATC/AD graphics requirements. This was not the case only a few years ago. Today any RISC based workstation can perform a range scale and offset faster than any custom AT/AD workstation of the past. The final trend is in the area of software tools and standards. With the use of a software tool, hundreds of thousands of lines of code are bypassed with 10,000-20,000 lines of data definitions. The power of such a tool can be used for such actions as changing how the track data is hooked and where hooked data is displayed. Such modifications require only the change of data definitions which are used to define the look and feel of the display, These data definitions control track symbology, track content, display lists, display maps, and even functions such as intercept calculations.

Further, housing a software tool on standards such as UNIX, X-Windows, and GKS permits the MMI design to be vendor independent. This vendor independence was shown at E-Systems when the same MMI design was hosted on a Tektronics XD88/11 (discontinued), SUN SPARC2, DEC 5000, and DEC 3100. The changing platforms were the results of hardware availability. This vendor independence will permit the lab to "float" with commercial technology which will allow for graceful upgrades to the training lab environment.

HARDWARE EMULATION APPROACH VS. SOFTWARE SIMULATION APPROACH COST

Figure 4 contains a cost trade-off between the bezel/shelf approach and the software CRT based approach. There are three cost drivers when implementing a hardware based emulation of the bezels and shelves of an ATC/AD workstation.

The first cost driver is the display monitor and its associated graphics generator. Many complex ATC/AD workstations such as the AAS sector suite are based on the Sony 20" × 20" 2048 × 2048 monitor. This is still a costly hardware item (figure 4, item 2). Switching to a more common monitor with smaller diameter and lower resolution (1280 × 1024) will significantly reduce costs ($75,000). For example, there is only a small difference in cost between a 19" diagonal monitor and a 23" - 25" diagonal monitor (~$5,000).

The second cost driver is the console enclosure itself. While supporting two program pursuits, E-Systems found that depending on the complexity of the console enclosure, the costs can range from $3,000 to $9,000. This cost is further increased if custom bezels and shelves are manufactured which use the actual switch panels in the fielded configuration. This cost includes the ordering and tracking of the parts and the development of a switch and "pot" scanner. The ordering and tracking of the actual panels with piece parts is no trivial task and required an automated inventory control program at the FAA

Cost Items	Hardware Emulation			Software Simulation		
	Man Mon	Eng ($50/Hr)	Purchase Cost	Man Mon	Eng ($50/Hr)	Purchase Cost
HARDWARE						
1. Commercial workstation hardware such as SUN SPARC2			$18,000			$18,000
2. Display (20" × 20" in OPS environment 19" trainer)			$75,000			$0
3. One console enclosure			$6,000			$0
4. Non-recurring engineering for bezels and shelves	6	48,000	$1,000	0	0	$0
SOFTWARE						
5. MMI design	0	0		1	8,000	$0
6. MMI implementation with prototyping tool	0	0		3	24,000	$0
7. MMI implementation in software code such as "C"	96	768,000		1	8,000	$0
8. Software licenses/display	0	0		0	0	$20,000
TOTALS - Quantity 1	102	$816,000	$100,000	5	$40,000	$38,000
TOTALS - Quantity 10	102	$816,000	$1,000,000	5	$40,000	$380,000
TOTALS - Quantity 20 (Site software licenses)	102	$816,000	$2,000,000	5	$40,000	$560,000

Ratio of hardware emulation to software simulation training platforms (Qty 1)	11
Ratio of hardware emulation to software simulation training platforms (Qty 10)	4
Ratio of hardware emulation to software simulation training platforms (Qty 20)	4

Figure 4. Cost Differences Between Hardware Emulation and Software Simulation Approaches

Technical Center for the ATCSF upgrade (this program was developed in BASIC on a Wang processor).

The third and largest cost driver is the approach for implementing the actual MMI design. If actual field software is selected, then actual field display hardware is required to support the training workstation. This hardware tends to be extremely expensive even if it is available. If there is an attempt to try to recode the display software on less costly hardware, then the non-recurring engineering costs are extremely high. Some organizations have concluded that the cost of developing the MMI using traditional software design and coding techniques can exceed 8 to 10 times the cost of using a prototyping tool such as InterMAPhics. E-Systems' internal experience has provided similar results.

However, this third cost driver could be eliminated with either approach if a rapid development/rapid prototyping software tool is selected.

COMPLEXITY

Figure 5 contains E-Systems' view of a complexity trade-off between the bezel/shelf approach and the software CRT based approach. There are three primary drivers in separating the complexity of these two approaches.

The first complexity driver is scheduling of training resources and executing the training session. In the hardware emulation approach, the laboratory resources are either idle or several users want access to the lab at the same time. This feast or famine characteristic in the lab leads to schedule conflicts and frustrated users, some of whom may have nothing to do with the training mission. This is a characteristic that plagues all valuable centralized resources. In addition, elements to establish a scenario are complex and include many personnel. For example, trying to emulate the actual operational floor requires voice communications and pilots on the other end flying airplanes in parallel with a predefined scenario. In the ATCSF and currently in the NSSF (reference 5), these pilots are seated at alphanumeric terminals and change the 12 or so airplanes they are "flying" based on voice communicated clearances from ATC controllers. Coordinating and running the script is no trivial task when the elements not only include the trainees but also other people in the loop to support the simulation of the actual system operations.

The second complexity driver is maintenance. With an attempt to emulate the operational floor of a fielded system, many other computer based components and subsystems are introduced. These can include trivial devices such as printers, to complex devices such as voice communications panels, or peripherally related items such as maintenance workstations. These components all form the lab and require maintenance to ensure proper operation of the training facility.

The third and perhaps most important complexity driver is the ability to change the training environment. All systems evolve, especially complex systems such as air traffic control or air defense. The training environments need to be able to support these systems no matter what the growth direction. The growth characteristics of two separate systems are never alike. The trainer must be more flexible than the operational environment to support a wider growth capability. Trying to emulate a fielded system with lower cost highly specialized hardware and software will most probably lead to less growth capability than the actual fielded system.

System Elements	Hardware Emulation	Software Simulation	Comments
1. MMI software in traditional code, such as "C"	1	N/A	The hardware emulation approach will probably be based on obsolete hardware
2. MMI implementation using prototyping tool	4	4	None
3. Scheduling of training resources	1	4	Anyone can "fire up" a standard desktop workstation
4. Executing training session	1	3 to 4	4 – Desktop setting, 3 – lab setting
5. Maintaining training laboratory	1	3	The software emulation approach is based on commercial products and less subsystems
6. Developing training scenarios	2	3 to 4	4 – Desktop setting, 3 – lab setting
7. Concurrency (Ability to change with operational environment changes)	1	4	The software emulation approach uses commercial hardware and commercial rapid prototyping software

4 – Very simple
3 – Simple
2 – Complex
1 – Very Complex

Figure 5. Complexity of Hardware Emulation and Software Simulation Approaches

In summary, the most significant issue is that the hardware emulation approach tries to emulate the operational floor, whereas the software simulation approach emulates the workstation functionality and simulates the interactions of the operational floor when desired. This is a significant difference and translates into a more manageable and more flexible training environment in the software simulation approach.

OPERATIONAL EFFICIENCY

Figure 6 contains E-Systems' view of an operational efficiency trade-off between the bezel/shelf approach and the software CRT based approach. There are three areas that are primary drivers in separating the operational efficiency of these two approaches.

The first operational efficiency driver is control room operations. The hardware emulation is the most effective approach for duplicating the operational floor. The software simulation approach can be configured in a laboratory setting with participants taking on the role of controller and pilot, but then the lab starts to take on some of the characteristics of the hardware emulation approach, especially from a complexity point of view as previously discussed. However, if the intent is to provide a training facility for a program such as AAS, which is using CRT displayed variable and fixed function keys, then the hardware emulation approach does not even apply.

The second operational efficiency driver is the ability to support multiple environments. Since the software simulation approach emulates only the functionality of a field workstation and it can be easily modified using a commercially available rapid development tool/rapid prototyping tool, it can easily be modified to support multiple environments. In the case of ATC, the current En Route workstations, TRACON workstations, Tower workstations, and the AAS workstations can be easily simulated with the functionality fully emulated.

The third operational efficiency driver is the availability of the trainers. Because of relatively small size and low cost, the software simulation approach trainers can be provided on the desktop, in a laboratory setting, or both. In a desktop application, the student or instructor can start, stop, or select a different scenario at will without impacting other students. This is enormous flexibility and permits the students to learn at their pace.

In summary, the primary difference between the two approaches is that the software simulation approach gives up the capability to physically emulate a workstation and the resulting operational floor for a trainer that is more available and tailored to the

Operational Efficiency	Hardware Emulation	Software Simulation	Comments
1. Console functions	4	4	None
2. Control room interactions	4	3	A lab is still possible with Software Simulation
3. Display surface (location of various items)	4	3	The hardware emulation approach may not even apply if system is based on CRT control
4. Field of view and reach	4	1	The hardware emulation approach may not even apply if system is based on CRT control
5. Tactile feedback	4	2	The hardware emulation approach may not even apply if system is based on CRT control
6. Operator response time performance	4	3	Reach is probably not a factor
7. Availability of training platforms/SIM facility	2	4	Lower cost means more platforms
8. Single operator dedicated trainer environment	1	4	Desktop possible
9. Student training time	2	4	Lower cost means more platforms
10. Student tailored lessons	1	4	Desktop environments
11. Ability to support multiple OPS environments (En Route, TRACON, Tower, etc.)	1	4	Provided by commercial rapid prototyping tool and commercial hardware

4 - Excellent
3 - Good
2 - Poor
1 - No Capability

Figure 6. Operational Efficiency Training Trade-off Between Hardware Emulation and Software Simulation Approaches

actual students. This is analogous to trading in the old batch computer room operations for the new interactive computer network with a terminal and now a PC in every office. Turnaround time is reduced, more flexible services are provided, and the resources are more accessible. The training lab is no longer providing system support or system study services.

CONCLUSIONS

The advantages and disadvantages of these two approaches are summarized in figure 7. The primary characteristic of the hardware emulation approach is that a traditional laboratory will be established where all the training displays are collocated in one facility that will duplicate the actual field environment. The primary characteristic of the software simulation approach is portability and low cost which can be used to establish relatively large numbers of desktop trainers.

History has shown that due to every day operations and cost constraints, traditional training labs have a difficult time of supporting even the functionality of ATC/AD workstations. With the use of software based CRT bezels and shelves, a RISC workstation, and a software tool, the functionality of the ATC/AD training workstation can be provided in a CRT based workstation. Further, the resulting low cost can provide up to four times more training time than with a training lab that attempts to emulate a field operational facility. In either approach, the development of the trainee's perception of the look and feel of the environment will be provided when the trainee arrives at the actual operational facility with its "unique" look and feel based on its particular operational characteristics.

REFERENCES

1. Ralph E. Whitney and Thomas B. Allen, "Selecting a Graphical Interface for Air Traffic Control Radar Simulation," 1990 ITSC Proceedings, pp. 150-156.
2. Cynthia Hubbell, "The Benefits of Desk Top Rapid Prototyping," 1990 ITSC Proceedings, pp. 313-319.
3. System Segment Specification for the FAA System Support Computer Complex (SSCC), 1986.
4. System Segment Specification for the FAA Research and Development Computer Complex (RDCC), 1986.
5. K. E. Ufelman, "Flying 800 Planes Simultaneously via Simulation," Photonics Spectra, April 1991, pp. 108-110.

ACKNOWLEDGEMENTS

Ron Johnson for his diligent review and excellent comments and Renatta Schmit for her guidance.

ABOUT THE AUTHOR

Walt Sobkiw is currently an independent consultant. Previously he was with the Advanced Engineering Group of E-Systems, Inc., ECI Division as a Project Engineer for two Air Force Air Space Management Systems. Prior to joining E-Systems, Mr. Sobkiw was at Hughes Aircraft Corporation for five years where he worked on developing the Federal Aviation Administration's Advanced Automation System (AAS). He developed the original AAS architectures, and defined the Air Traffic Control Workstation Common Console concept. He also supported an internal IRAD to develop a generic Air Traffic Control/Air Defense workstation.

Mr. Sobkiw was formerly at the FAA Technical Center where he supported the upgrade of the Air Traffic Control Simulation Facility (ATCSF) and was responsible for the development of the generic En Route and TRACON workstations in that lab. Mr. Sobkiw holds a bachelor's degree in Electrical Engineering from Drexel University and is a member of IEEE, AFCEA, and ATCA.

Approach	Advantages	Disadvantages
Hardware Emulation	• Emulates real workstation • Provides same tactile feedback • Provides same operator field of view • Forces the same operator movement	• High cost • Specifically tailored to one system • No longer applicable to many next generation systems • Limited student availability
Software Simulation	• Low cost • Potential dedicated trainer for each student • Emulates functionality • Emulates physical characteristics of new generation workstations • Can support training for various systems and versions of systems	• Simulates not emulates physical characteristics of older workstations

Figure 7. Summary of Advantages and Disadvantages of Hardware Emulation and Software Simulation Approaches

RECONFIGURABLE SIMULATORS FOR SPECIAL OPERATIONS FORCES MISSION REHEARSAL

Richard Vestewig
Perceptronics, Inc.
Woodland Hills, CA

Carl Bergsneider
Loral Defense Systems
Akron, OH

Capt Scott Richardson
ASD/YWSA
Wright Patterson AFB, OH

ABSTRACT

The Special Operations Forces Aircrew Training System (SOF ATS) is facing the challenge of providing Mission Rehearsal capability to SOF crews, for any complement of aircraft that the mission demands. To meet this, SOF ATS is developing reconfigurable Mission Rehearsal Devices with the capability, through software and hardware changeout and modification, of providing simulators which look and act like any SOF aircraft. The development of the reconfigurable MRDs required extensive analysis of aircraft commonality, mission requirements, and a unique concept of virtual displays coupled with working physical components. The operational context for mission rehearsal in the SOF community is discussed. Finally, research questions and how they may be answered in the SOF ATS mission rehearsal suite are addressed

Introduction

The Special Operations Forces (SOF) is growing in importance as a United States military component. Due largely to changes in the world situation, and typified by the Persian Gulf War, the need for a capable, mobile, highly coordinated force using assets from each service has become evident. Critical to the effectiveness of SOF is training, not only in the duty positions of each unit, but also in teamwork and coordination. In addition, mission training requires that essential mission critical tasks must be practiced, critiqued, and improved before the mission actually takes place.

The goal of the Special Operations Forces Aircrew Training System (SOF ATS), awarded to Loral Defense Systems and its subcontractors by the Air Force, is to provide training and mission rehearsal capability for all crew members for each of seven aircraft: the MC-130E COMBAT TALON I and MC-130H COMBAT TALON II, the AC-130H and AC-130U gunships, the HC-130N/P tanker, the MH-53J helicopter, and the MH-60G helicopter. The training segment covers all components of crew member training, from initial mission qualification through continuation and upgrade training. Mission rehearsal, however, has the unique role of preparing crew members for actual mission performance. Mission rehearsal is provided in part by specially designed and constructed simulators stressing crew coordination and communication. These simulators, the Mission Rehearsal Devices (MRDs), are in addition to the Weapon System Trainers (WSTs), which provide simulator training on all aspects of aircraft operation.

Given the various nature of SOF missions, the MRDs need to be extremely flexible. In particular, they must allow rehearsing missions in any part of the world that SOF is required; the mission data base must be produced within 48 hours, and the MRDs must be able to represent any SOF aircraft, so that the mission can be rehearsed with the correct aircraft complement This paper describes the concept, design and status of the reconfigurable MRDs being developed for SOF ATS Each of the major components of reconfiguration physical configuration of the cockpits and operational panels, and the software reconfiguration of the virtual panel concept – have been successfully demonstrated and are under way for on time delivery.

Advantages of Reconfigurability

A reasonable question is why reconfigurability is desirable, especially since fully capable Weapon System Trainers (WSTs) are being developed for each of the SOF aircraft. The most important reason is the diverse range of the SOF missions. The MRDs must support an extensive list of tasks and missions including, for example, Airland, Airdrop, Reconnaissance, Search and Rescue. All of these must also be performed under Night Vision Goggle (NVG) conditions. However, although each of these can be single ship missions, they may also require a number of aircraft of any type working together. A dedicated device simulating a single aircraft can not meet the mission flexibility necessary to practice diverse missions. However, if the simulator consists of a basic core with reconfiguration modules, then any SOF aircraft can be obtained simply by changing the modules.

Designing the simulators to be reconfigurable also has cost advantages, by exploiting hardware and software commonality. For example, C-130 MRDs share physical commonality of all C-130 aircraft, since the same aircraft platform is used for all. Reconfiguration of one variant of C-130 to another requires changing only those internal physical components and software unique to the aircraft. There are also physical and functional similarities between the SOF fixed wing aircraft and the rotary wing aircraft. including seat position, placement and type of control panels, and front instrument placement.

Software commonality and interoperability is achieved by two means. First, the MRDs use as much as possible the same software as the Weapon System Trainers, although, as described below, not all WST systems are active in the MRDs. Second, the C-130's share much the same flight dynamics, so that only minor modifications are needed to the software to simulate the C-130 characteristics. However, fixed wing and rotary wing software is unique, and requires separate development.

These commonalities were exploited to generate common hardware modules that greatly simplified the design of the reconfigurable MRDs. Moreover, a unique concept for depicting instruments and controls on the flight instrument console and in the crew stations, using common software reconfigurable CRTs in conjunction with overlays containing actual switches, led to a common design for all flight instrument consoles and for all crew stations.

Philosophy of Mission Rehearsal and Reconfiguration

The Mission Rehearsal Device is a unique concept used in SOF ATS. It works together with Weapon System Trainers and classroom instruction to ensure that the crewman can perform all functions required in his aircraft and also practice coordination and communication skills. With both of these capabilities in SOF

ATS, Mission Rehearsal can focus on mission critical skills and not duplicate capability provided by qualification training and the Weapon System Trainers.

The overarching principle governing the development of the MRDs is that mission rehearsal is not mission qualification training. The crewmen undergoing mission rehearsal is assumed to be mission qualified. Consequently, not all controls, displays, or sensors active in the aircraft need be functional in the MRD. The MRD contains only components critical to mission performance. Components not deemed critical to mission performance are depicted by pictures, or with reduced capability from the actual aircraft (e.g., less than complete radio channel capacity).

The MRD is a reduced functionality device, but not a low fidelity device. Rather, the principle of "selective fidelity," used successfully by Perceptronics in the development of the SIMNET ground and air vehicle simulators for DARPA, was employed in the development of the MRDs. The functionality provided gives sufficient cues and interaction for successful mission performance. Functional characteristics of critical components, such as the radars, are provided with high fidelity. However, switches will not necessarily be exact replicas of actual aircraft switches, although their actuation will result in the correct outcome in the simulation. This simulator implementation illustrates the difference between traditional transfer of training requirements and mission training. Unlike physical and procedural transfer from a training device to the actual equipment, transfer to mission rehearsal focuses on cognitive transfer to the skills required for crew and multi-aircraft coordination in a mission scenario. Thus, the functional fidelity necessary for cognitive performance is preserved, although the physical manifestations of those functions may be somewhat different from the actual aircraft.

The mission rehearsal requirements and the assumption of a qualified crew member are the major drivers in developing the concept for MRD reconfigurability. In particular, compromises to full fidelity were made to meet reconfigurability requirements, but not at the expense of mission rehearsal and performance practice. For example, some modification in the overall physical configuration of the MRD was made so that common components could serve each of the SOF aircraft. These components included the seats, support floors, and common support structures for the overhead consoles and the center consoles. Using common support structures means that only specific panel component need be reconfigured.

Some physical components critical to mission performance were not modified at all. Most important of these are the window frames. Pilots use the position of the window frames relative to the out-the-window view for positioning the aircraft, for areas such as landing or formation flying. Consequently, in the MRD these were modeled exactly so that the crew member's view out into the visual scene was an extremely high fidelity replica of the view out the aircraft windows.

Reconfigurable MRD Description

The SOF ATS reconfigurable Mission Rehearsal Device consists of two major components:

- Physical structure The physical structure of the MRD houses all of the components and provides the overall flight deck and crew station environment for the crew members.

- Functional structure. The functional structure consists of the working components for each of the aircraft panels and the crew stations It also consists of a unique devel opment of virtual flight instrument and crew consoles interacting with actual physical components

Both of these are reconfigurable to any of the SOF aircraft using modular kits.

Physical Layout The MRD flight station is a fixed installation. It consists of a floor structure, front instrument console installation, relevant console structures, and a U-shaped "strongback" which supports the overhead consoles with a fixed tine, serves as the connection point for the window frame structures, and is the interface between the flight deck and the crew station enclosure. Attached to the back is an additional structure which houses the crew stations for the aircraft (for the rotary wing aircraft, the crew stations are not used). The entire structure is inside a partial dome onto which the relevant visual images are projected. The MRD is fixed on a platform under which is placed relevant electronics and storage.

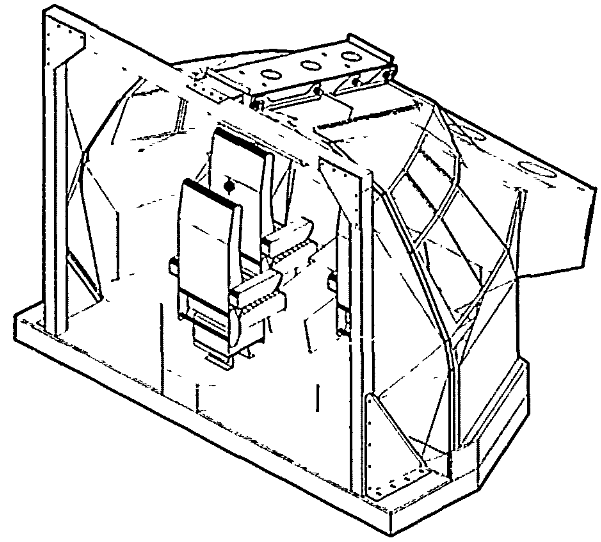


Figure 1. Reconfigurable MRD Flight Station

The flight deck has been designed to be easily reconfigurable. In particular, the window framing system can be removed and replaced, each of the panels can be replaced (when reconfiguring from a fixed wing to rotary wing aircraft, some of the structural consoles are replaced also), and primary controls are replaced. At the same time, simulation software is also being replaced to represent the correct aircraft.

The front instrument console is a fixed installation consisting of three large CRTs, and is not physically reconfigured. These are software reconfigured to represent the target aircraft using a virtual display methodology (described below).

The flight deck contains seats for the pilot, co-pilot and flight engineer. It also contains all the switches and controls necessary for flying the aircraft and carrying out the mission, including primary flight controls (yoke, pedals, cyclic and collective for the rotary wing aircraft), overhead and center consoles, and side consoles (fixed wing only). The flight deck contains two floor structures which house the control loading system and provide interfaces to the overall raised support.

Enclosing the flight deck is a window framing system to replicate the out-the-window view of the aircraft. Attached to the window frame is a light tight shield between the window frame and the crew station enclosure. The window frames are modeled closely on the actual aircraft, and are replaced when reconfiguring from a fixed wing to a rotary wing aircraft.

Development of the basic concept for the reconfigurable MRD layout relied heavily on the concept of commonality and human factors. Although aircraft have obvious physical differences, the anthropometry of each must be such that humans can fly them. Consequently, although the physical envelope of the aircraft differ markedly – witness the C-130 and any rotary wing helicopter – the flight deck, controls, displays, and out the window views are extremely similar. A significant part of the concept development was extensive measurements and photos within the cockpits of all of the SOF aircraft – the C-130, the MH 53J, and the MH 60G. The measurements were examined to determine as much commonality between the physical layouts of

the cockpit so the MRD could use common components without compromising training and mission rehearsal performance effectiveness.

Significant commonality was found in the measurements. Some examples are:

- All aircraft have overhead and center consoles. All C-130 variants have side consoles and side shelves in the same positions.
- Center to center pilot-copilot seat separation for the three aircraft varied plus/minus 2 inches.
- Angle of the center console varied plus/minus 3 degrees
- Width of the center console varied plus/minus 1-1/2 inches

This information was used to develop an overall MRD configuration using common structure and support components. The MRD seats are fixed for all three aircraft (this design also allows a common eyepoint for the visual system focus). A single support structure for the center console is used for all aircraft. A common overhead support structure is used to support all overhead consoles.

The design concept distinguished between the consoles structures and the functionality on the consoles The console structures support the panels which house the active switches and controls, and provide electrical connect points and interface to the I/O system. Thus, the console structures are common to all C-130 MRDs, but the panels on the structures are different depending on the aircraft configuration and mission. When reconfiguring to a rotary wing aircraft, the console support structures are replaced with the appropriate configurations for that aircraft.

Crew stations are a common design for all the SOF aircraft. These include the two crew stations on the MC-130E and HC-130N/P, and the single crew station on the MC-130H. This single design accommodates all relevant functionality for the specific crew stations, and is reconfigured to virtual displays (described below).

Functional configuration. The central component of mission rehearsal is the functionality provided by the MRD. The MRD is a reduced functionality device which provides mission critical functions. However, the mission critical functions are portrayed with sufficient fidelity so the qualified crew member has the necessary physical, functional, and cognitive cues to perform his mission in coordination with his own crew and other aircraft crews.

Functionality is specific to each aircraft, although, especially for the C-130 variants, there is much commonality. In general, however, reconfiguration between aircraft requires changeout and replacement of panels which contain the functionality, although the overall support structures may remain.

If not all functions are provided, which are, and to what degree of fidelity? Necessary functions were determined by an extensive analysis cycle which began in Phase I of the SOF ATS program and has continued until the Preliminary Design Review (PDR) of the Phase II development. Subject matter experts provided input on the mission criticality of all components of the SOF aircraft. Initially, the component was determined to be active or not. Those not necessary for mission rehearsal were to be portrayed by some static representation. Each active component was classified for required fidelity as either *Replicated* (must function and feel exactly like the actual aircraft component), *Depicted* (must function like the actual component, but may vary in feel or placement), or *Inert* (must provide physical feedback, but need not function) The latter category was used for non critical components used by the crew member for physical location of critical components, especially those he locates by blind reach.

Overall, approximately 40% of the components are required to be either replicated or depicted. Replicated components consist mostly of primary control heads, such as the yokes and pedals. These are provided by actual aircraft or simulator components, identical to those in the Weapon System Trainers. Depicted components provide all of the required functionality, but typically use commercial switches and dials rather than actual aircraft components.

The functional development described above is followed for all of the overhead, center, and side consoles. Reconfiguration consists mostly of panel replacement to reflect the particular aircraft. However, functional depiction and reconfiguration for the primary displays - the flight instrument console and the crew station consoles – provided a singular challenge. Not only do these differ significantly between aircraft, but they are the primary information displays and mission critical stations and cannot be compromised. The challenge for the reconfigurable MRD concept is to provide easy reconfiguration of these primary information components, but without the physical and functional commonality present in the physical layout and the consoles. Loral has developed a unique solution to this problem, using a combination of virtual displays and actual functioning controls to meet both ease of reconfiguration and correct, mission critical switch action.

Two approaches were investigated. The first approach was to determine if enough commonality existed between the instrument panels and crew stations of the aircraft to make feasible use of generic instrumentation pattern. Analysis quickly determined that the mix of instrument panel sizes, panel placement relative to the crew member, and mix of instruments (altimeters, compass dials, oil pressure dials, multi-function CRTs, etc.) would require an instrument panel especially designed for that model of aircraft. This approach greatly complicates reconfigurability, since an entire instrument panel would need to be replaced. Overall, this approach would be time intensive and extremely costly, since the savings provided by commonality would be minimal.

The second approach was to determine if "virtual" instrument panels created through software graphics would be acceptable. This approach uses a single soft panel CRT complement reconfigured with software to simulate the particular aircraft. Switches and dials are actuated by a touch screen representation to simulate the switch movement. The disadvantage to this approach is that the virtual flight instrument and crew station panels with touch panels do not provide three dimensional actuation and feedback, so that task performance may be hindered.

The chosen solution is a hybrid where a large part of the graphics software reconfigurability is retained along with certain hardware control panels affixed to the CRTs. In particular, a limited number of functions are provided as actual switches and keyboards on overlays to the CRT virtual display. These interact appropriately with the information depicted on the virtual displays. The switches embedded in the overlays are those for which tactile feedback is critical, and for commonly used controls. Other functions are provided by the touch panels, which are permanently fixed to the CRTs. Thus, reconfiguration is accomplished by software modification of the virtual displays and the touch-screen activation areas, and replacement of the overlay containing actual switches with another appropriate to the target aircraft.

The virtual display concept, combined with touchscreens and overlays, requires significant human factors analysis to design a system which provides all mission critical functions but does not differ enough from the actual configuration to impede task and mission performance. This analysis is particularly important in the modification of instrument placement on the actual flight instrument console or crew station to meet the layout of the virtual displays (three CRTs for the flight instrument console, and two banks of three CRTs for the crew stations). Instrument placement on the CRTs to mimic actual placement on the MC 130E was accomplished as follows. In the actual aircraft, pilot and copilot flight instruments are placed appropriately in front of the crew member, with various engine instrument indicators placed in between. This depiction places all of the engine indicators on the middle CRT, with the pilot and copilot instruments in the left and right CRT respectively. Figure 2 illustrates the nominal placement of the MC-130E flight instruments on the three CRTs making up the virtual display. This arrangement preserves the essential nature of the instruments for both crew members, although some modification

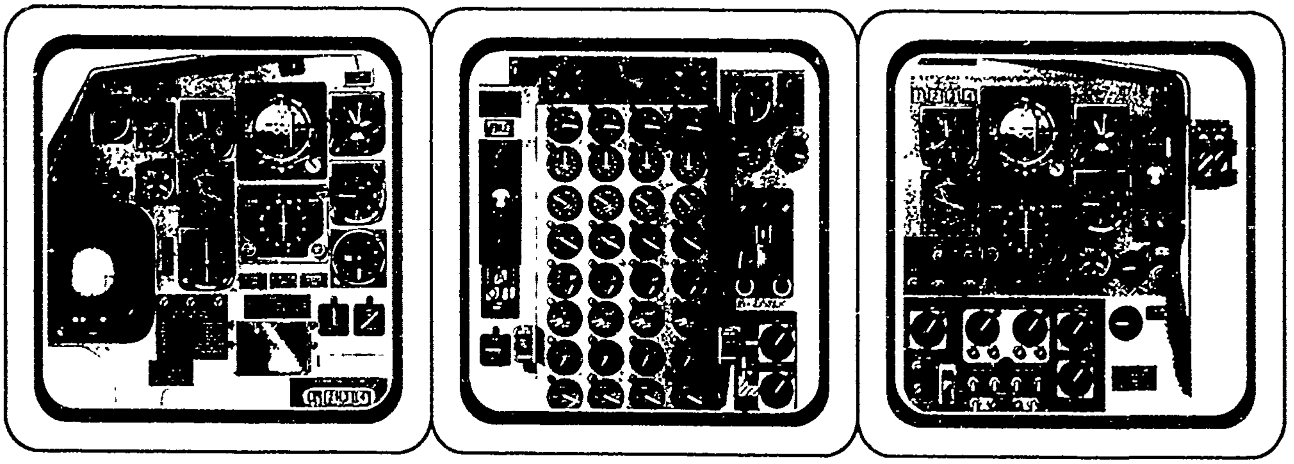


Figure 2. Nominal Placement of MC-130E Instruments on Front Console CRTs

of size and instrument position have been made to accommodate the CRT limitations. The analysis to place the instruments on the virtual display is underway for all the SOF ATS aircraft.

Implementation of Virtual Displays for Reconfigurable MRDs

The reconfigurable flight instrument console and crew station concept is graphics intensive simulation whereby large, high resolution monitors are used in place of replicated instrumentation in crew station panels. In order to maintain high fidelity, a specially developed photodigitizing process was used. The process is as follows:

1. Photographs of the cockpit instrumentation are taken with special consideration for sharp images of each instrument, a minimum of glare on the face of each instrument, and a minimum of off-axis viewing for each instrument. In addition, where lighting condition is a factor, various photos of each instrument can be taken (bright daylight, normal, dusk, dark night, etc.) to be able to present that image when appropriate.

2. The photographs are scanned to produce digital image files. Scanned images of each instrument are sized to produce a life-size image of the instrument on the simulator screen.

3. The digital image files are edited to remove moving instrument parts such as dial needles, alphanumeric tumbler legends, light bulb bezels, and switches. After editing, the image file contents that remain are the static presentations of the instrumentation that only need to be loaded once (no updates required) when it comes to displaying the cockpit instrument panels.

4. The moving parts of the instruments (dial needles, alphanumeric tumbler legends, rotary dial faces, etc.) are then caligraphically created using a commercial graphics package and are stored in a callable object library. In some cases, the same object can be used on several different instrument dials (as is the case of some dial needles) and in some cases, a piece of the file prior to editing can be utilized to create the object (as is the case for light bezels where one light is on and the other is off, or in the case of switches where one switch is up and the other is down so that an example of each is available for use).

5. Depending upon which cockpit version is to be presented to the crew member, the appropriate static images are sent to the graphics engine where they are loaded onto the video bit planes designated as the background image video memory planes. The more bit planes that are allocated for use, the richer the color mix. Not all the bit planes can be used for the background static images since one or more are required for the object animation overlay.

6. Depending upon which cockpit version is presented to the crew member the appropriate graphics objects (dial needles, etc.) are drawn on the screen at their correct position and updated as required to provide a perception of motion. Since most of the display is static, a high update rate can be achieved with the objects thus allowing for a very smooth motion perception.

7. If there is a cockpit CRT in the instrument panel, a video "window" is created, live external video is piped in, and the graphics engine digitizes, sizes, and draws the live video image in the CRT area.

8. A touch-screen overlay is placed on the surface of the display CRT so the crew member can interact with the cockpit instruments albeit with a reduced sense of realism due to the inability to feel the actual contours of the instrument controls. Host software determines the effective touchsense area and position, and mechanizes the instrument control seen on the display appropriately. Such cockpit controls as switches, rotary knobs, handles, and pushbuttons are made interactive using this method.

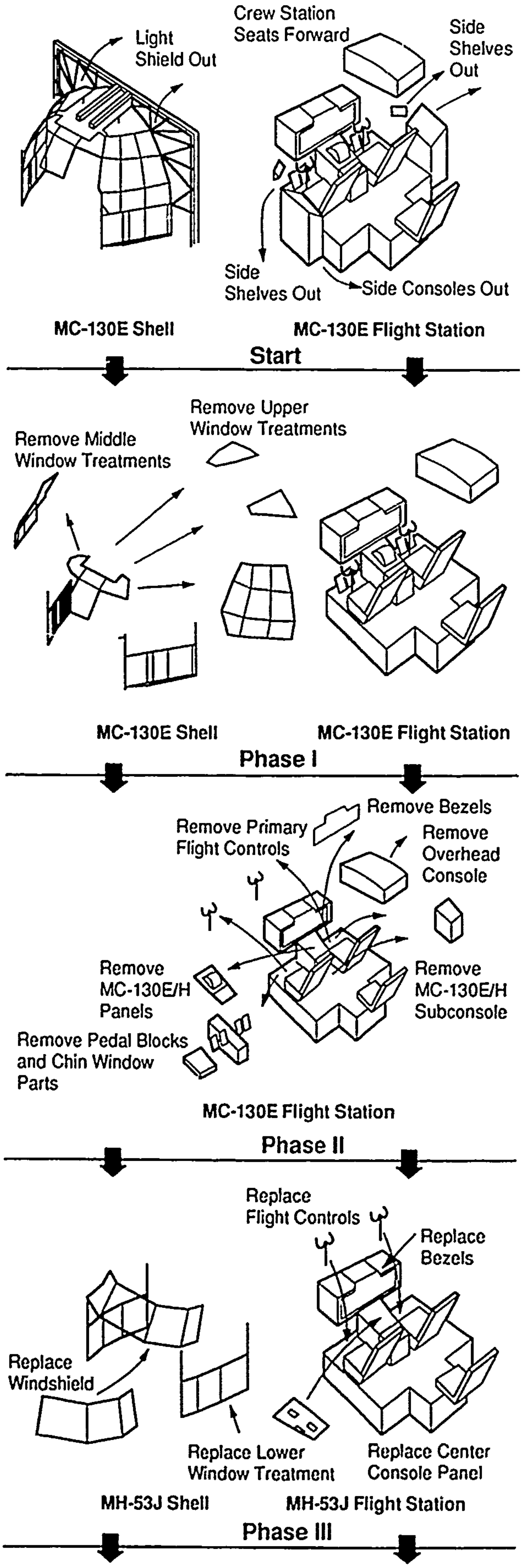
Reconfiguring the Mission Rehearsal Device

The SOF ATS mission rehearsal specification requires that reconfiguration of the MRDs be accomplished in 2 hours or less. This is a stringent criterion, and meeting it relies heavily on component commonality and changeout of only necessary components.

Reconfiguring the MRD requires two parallel task sequences, the physical reconfiguration and the functional reconfiguration. Physical reconfiguration is accomplished using simple procedures which replace only those components different between the aircraft. Each of the steps requires either an individual or a two man team.

Overall, the physical reconfiguration of the MRD is a 24 step process, broken down into four phases. The most complicated is reconfiguration from a fixed to a rotary wing aircraft, since there are significant differences in physical layout. These steps are illustrated in Figure 3.

The first phase dismantles the window treatments, removing the windshield, the side window treatments, and the light shield. This leaves intact the flight station, with the controls, center console, and overhead console (the overhead console is supported by a fixed "strongback" structure). In the second phase, the primary flight controls, overhead console, and center console panels are removed.



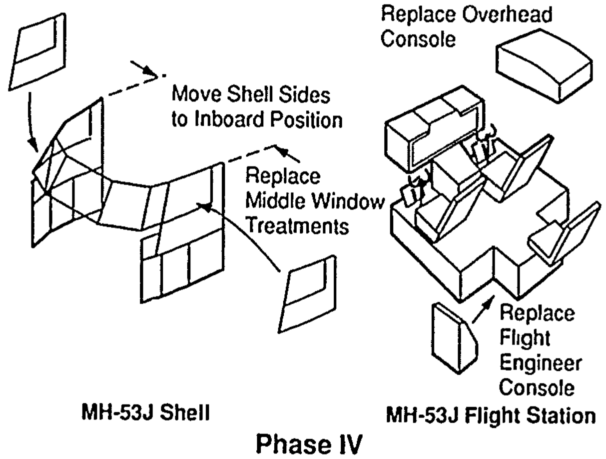


Figure 3. Reconfiguration Steps for SOF ATS MRDs

Phase 3 begins the process of reconfiguring the MRD to a rotary wing aircraft. Relevant primary flight controls are installed to a common control loading system, and the center console is fitted with the rotary wing instrument cluster. The unique flight engineer's console is installed. The rotary wing overhead console is then installed on the strongback In phase 4, the correct out the window view is achieved by replacing the fixed wing window frames with the rotary wing frames. In addition, the overlays are removed and replaced on the fixed front instrument console. This reconfiguration is performed as the entire software component is replaced, to reflect the correct flight profiles, instrument I/O, and virtual display.

When crew stations are reconfigured, the software controlling the crew stations is replaced, in conjunction with the virtual displays, the overlays, and touch panel implementation.

When an aircraft is reconfigured from a fixed wing to another fixed wing aircraft, such as between the MC-130E and MC-130H, the reconfiguration process is considerably simplified. No replacement of the window treatments nor of the primary flight controls is needed. The major reconfiguration is in the virtual displays and overlays, owing to the differences in information display and control between the glass cockpit MC-130H and the traditional round dial MC-130E, and the unique crew stations for the aircraft.

SOF Training and Mission Rehearsal Operational Concepts

The Mission Rehearsal Devices for SOF ATS will be part of an integrated training system for all of the SOF crews in the Air Force component of the Special Operations Forces. When the overall complement of devices is fielded, there will be 5 reconfigurable MRDs, one each for the MC-130E/H, HC-130N/P, MH-53J, and MH-60G. Enough reconfiguration kits will be procured to allow any complement of aircraft to take part in the mission rehearsal. In addition, AC-130H/U WSTs will also take part in mission rehearsal. All of these devices will be colocated at Hurlburt Field, Florida.

SOF is a close knit community of specially qualified personnel in unique organizations. These organizations are structured and administered by a service Special Operations Command, such as Air Force Special Operations Command (AFSOC), who are military service components of United States Special Operations Command (USSOCOM).

AFSOC, the Air Force component of SOF, was established as an Air Force Major Command (MAJCOM) in 1990. Previously AFSOC was dual roled as 23 Air Force under Military Airlift Command (MAC) in addition to AFSOC. AFSOC is made up of Special Operations Wings (SOW) with assigned squadrons which provide uniquely configured fixed wing, vertical lift, and fire-support aircraft for SOF. AFSOC and the 1 SOW are located at Hurlburt Field, Fl.

The SOF ATS will, when fully developed and deployed, provide schoolhouse training for the MC-130H Combat Talon II, MC-130E Combat Talon I, MH-53J Pave Low, MH-60G Pave Hawk, and HC-130P/N Combat Shadow aircraft at Kirtland AFB, NM. The SOF ATS will also provide schoolhouse training for the AC-130H and AC-130U Spectre Gunships in addition to the integrated combat mission rehearsal capability at Hurlburt Field, Fl. In addition, SOF ATS will provide continuation training for operational locations world-wide.

The mission objective for special operations activities are normally defined through the response to a developing situation that would unfavorably impact on US Government interests. The definition of a specific mission, for potential assignment to SOF, could begin with either a theater Commander request or from NCA monitoring of events world-wide. Once a mission option investigation has been initiated, the Joint Chiefs of Staff must set the mission constraints to bound the activity. A preliminary Concept of Operations to achieve the unique and specific goals of the potential operation will then be developed. SOF missions will be on of two types, time sensitive and non-time sensitive.

Time sensitive missions are those "crisis" missions in response to fast breaking threat situations. There is limited time to gather intelligence information for the specific mission area, so maximum use of archival material and current on-hand intelligence information is required.

Non-time sensitive missions are those developed as "deliberate" or contingency planned missions. Deliberate planning normally is conducted as a hedge against potential threats that may interfere with US interests in any theater of operation. Generally, the "Hot Spot" is known, and there is common knowledge of US interests. A deliberate intelligence collection plan is formulated and submitted for prioritization within the intelligence community. This type of planning can either support war plans or can support SOF unique plans.

Mission rehearsal is the means by which various elements of a mission can achieve some measure of practice and proficiency using the planned mission scenario for a special operation mission. Successful mission rehearsal is the key element for force multiplication through minimizing the potential for problems during execution. Mission rehearsal within SOF is currently accomplished in actual aircraft using mockups (when practical), checklist rehearsal, and by simulations. Rehearsal in aircraft poses a number of Operational Security (OPSEC) problems.

The SOF ATS combat mission rehearsal capability will provide SOF the ability to get aircraft-like realistic rehearsal without the OPSEC problems associated with using actual aircraft. The SOF ATS mission rehearsal system will respond to the time-sensitive missions in a secure environment. The Mission Rehearsal Devices (MRDs) representing each of the Air Force SOF aircraft will be linked in a computer network in so that a secure rehearsal can be accomplished.

SOF ATS mission rehearsal is keyed to visualization (out-the-window, night vision goggle, radar, electronic warfare, and other sensors providing a visual display to aircrews) of a simulated fly through of terrain simulating the actual mission area terrain. This visualization of the mission area will not only prepare aircrews participating in the missions by rehearsing them and letting them gain proficiency, but it will also be a key ingredient in the commander's decision making process. With the expected fidelity of mission rehearsal in the SOF ATS, the mission commander will be able to witness the mission unfold to determine if the mission plan, as executed in the rehearsal, is a viable plan with an acceptable probability of achieving mission objectives. Once these are determined, the mission commander can then properly advise the NCA regarding the mission preparation and readiness to execute.

Conclusion and Next Steps

Mission rehearsal is a concept growing in importance as the world situation changes. Key to its success is the flexibility to provide realistic mission rehearsal for any contingency, planned or unplanned. In addition, any conceptual methodology which reduces the overall cost of mission rehearsal as well as increases its effectiveness will pay significant dividends both in readiness of the crews, ease of use, and frequency of use. The SOF community can especially benefit from this capability, and SOF ATS will provide it for them.

Simulator-based mission rehearsal is a relatively new concept in an area that has traditionally relied on checklist and limited team practice in aircraft or simulators. Expanding this to include the design and development challenges of easy, fast hardware and software reconfigurability without compromising effectiveness greatly increases the use of the simulators for any mission. Although these developments, performed in conjunction with the Air Force ASD by Loral and its subcontractors, are well within the state of the art, these new developments also offer questions on how to use this new capability for optimal effectiveness.

Andrews, Nullmeyer, and Fuller (1990) describe some of the research issues in mission rehearsal that these new capabilities can address. They note that there is little research on mission rehearsal, but that the existing body of research in performance provides some insight in the types of learning that are involved in mission rehearsal. Clearly, though, a major lack in the research is the implications for performance of the assumption that the crewmen is mission qualified. Particularly, does being mission qualified in basic skills transfer optimally into the types of coordinated team behavior necessary for mission performance? Answering this question is hampered further by the lack of well established criteria for team performance in emerging situations.

The Mission Rehearsal capability that SOF ATS will provide can help answer important questions for improving the effectiveness of simulator-based mission rehearsal. As Andrews *et al.* describe, these include instructional features, feedback, and performance measurement. Mission rehearsal performance can also feed back into the mission qualification training, by pinpointing those areas in schoolhouse training especially applicable to effective mission performance. In addition, the use of the concept of selective fidelity provides fertile ground for knowledge in the development of reduced functionality mission rehearsal simulators for areas outside SOF ATS.

Overall, the success of the reconfigurable simulator for SOF ATS mission rehearsal will lead to greater readiness for the crews, greater assurance that the mission will be carried out successfully, and greater survivability for the SOF members.

REFERENCES

Andrews, D., Nullmeyer, R., & Fuller, J. Mission Rehearsal Behavioral Research Issues. *Proceedings of the 12th Interservice/Industry Training Systems Conference*, 1990, pp. 401-408.

ABOUT THE AUTHORS

Richard Vestewig is Vice President, Systems Training and Program Director, SOF ATS at Perceptronics in Woodland Hills, CA. Prior to joining Perceptronics, he was Section/Project Chief at the Honeywell Systems and Research Center, Minneapolis, MN, Research Psychologist at the AF Human Resources Laboratory, Wright Patterson AFB, and Associate Professor of Psychology at Wright State University, Dayton, OH. He holds a B.A. from Yale University and a Ph.D. in Psychology from the University of Michigan.

Carl Bergsneider is Principal Investigator for Research and Development in the Simulator and Trainer Engineering Department of Loral Defense Systems-Akron. He previously contributed to development of the F-15 Operational Flight Trainer and F-15E Weapon System Trainer projects at Loral in the weapons and armament systems modelling area. Prior to joining Loral, he spent six years at McDonnell Aircraft in the Flight Simulation Laboratories in St. Louis, MO. He holds a B.S. in Aerospace Engineering from the University of Arizona.

Scott Richardson is a Captain in the Air Force and Program Manager, Special Operations Forces Aircrew Training System Simulators. He is presently assigned to ASD responsible for acquiring an integrated training and combat mission rehearsal system for the Special Operations Forces. He was previously assigned to the Strategic Air Command for acquisition, configuration management and test of B1 B, B-52, and KC 135 weapon system trainers, and to Air Training Command for programming of the T-37/T-38 flight simulators. He holds a B.S. in Business Administration from Plymouth State College, Plymouth, NH.

Battlefield Smoke - A New Dimension in Networked Simulation

Rick D. Bess
Brian T. Soderberg
BBN Systems and Technologies Division
Bellevue, Washington

ABSTRACT

The use of atmospheric obscurants such as battlefield smoke to modern day tactics is critical. Recent military activity in the Middle East Gulf Conflict has highlighted the impact of reduced visibility on manned vehicle and smart weapon system effectiveness. Battlefield smoke is used for tactical cover and concealment, to silhouette targets, and to cause enemy disorientation and confusion. The simulation of this feature will ensure faithful and comprehensive tactical team training for armor, and airborne vehicles.

The technical challenges presented by the simulation of volumetric atmospheric obscurants have hindered prior implementation of battlefield smoke in tactical trainers. This paper considers technical limitations associated with simulation of visual effects of smoke using real-time computer image generation, as well as less obvious problems such as the effects of smoke on various sensors (e.g. thermal sensors). Further, emphasis is given to challenges associated with creating a consistent and realistic simulation of smoke for trainers that are networked together in a distributed simulation environment Recent advances in real-time computer image generation and simulation system technology can now be applied to solutions for simulating battlefield smoke.

This paper provides an overview of the issues associated with the visual simulation of atmospheric obscurants (e.g., battlefield smoke) in tactical team training First, it reviews the training requirements for atmospheric obscurants in training systems by providing background on the tactical significance. Secondly, the problems associated with simulating obscurants such as smoke in tactical trainers are discussed. Finally, solutions to these problems are proposed. Photographs and video tapes will be used to illustrate the benefits of proposed solutions.

Introduction

The ability to simulate armor combat via networked simulators has sparked the desire for further battlefield realism. In response to these evolving needs, innovative technologies are being developed such as visual simulation of 3-D volumetric atmospheric obscurants. Smoke, dust, and poor weather have been and continue to be utilized as part of military tactics. These degraded atmospheric conditions alter the view of the battlefield for soldiers, and weapon systems.

Computer based, networked training systems such as SIMNET [1] have demonstrated that team tactics can be effectively taught using simulators. SIMNET has also prompted insightful feedback for areas of improvement. The realistic portrayal of battlefield obscurants was noted as a deficiency of the system for tactical training. [2]

In this paper, the background of atmospheric obscurants associated with team tactics is covered to better understand the technical implementation of the visual simulation. The military weapon system details and tactical descriptions included in this paper are based on documented feedback from U.S. and German Army experts, military smoke munition manufacturers, and simulation system research and development projects. In the second part of the paper, a comprehensive examination of the technical challenges of visual simulation of volumetric atmospheric conditions is provided. This paper concludes with innovations in computer image generation technology that address these visual simulation needs.

Training Requirements for Smoke

For the purpose of discussion in this paper, the use of tactical smoke will be used as the primary example of volumetric atmospheric obscurants. Training requirements for team tactics that include smoke are uncovered by examining the military significance of smoke, the deployment methods, and the tactical maneuvers employed in combination with smoke. Effective training will prepare the soldier in the proper use of smoke, and solicit the predicted responses when affected by the obscured visual conditions.

Military Significance of Tactical Smoke

The military benefits of tactical smoke [3, 4] are summarized in Figure 1. These benefits represent why smoke is deliberately used to improve the soldiers effectiveness on the battlefield. In other situations, smoke may have potential disadvantages [3, 4] as listed in Figure 2. For example, if the soldier has superior fire control over the enemy, smoke deployment would not be preferred.

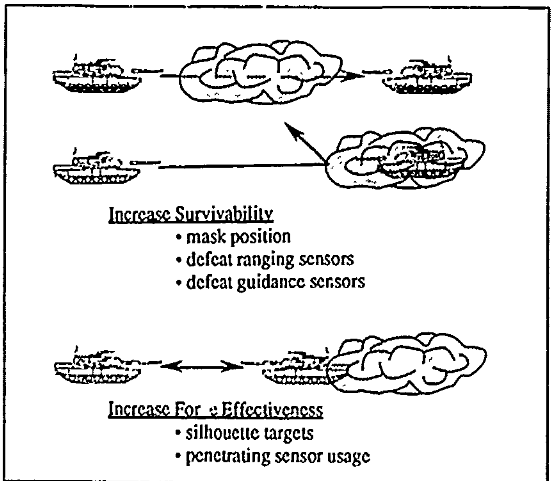


Figure 1 - Military benefits of tactical smoke.

Decrease Force Effectiveness
- Hampers movement coordination
- Degrades target acquisition & engagement
- Marks location to enemy
- Adverse psychological effects
  (isolation, disorientation)

Figure 2 - Military disadvantages of smoke.

Effect on Weapons Systems

The majority of Army weapon systems have at least one common trait: reliance on electromagnetic energy propagation through the atmosphere. Virtually all viewing systems, sensors, and laser systems are degraded in some manner by obscurants, as shown in Figure 3. [5]

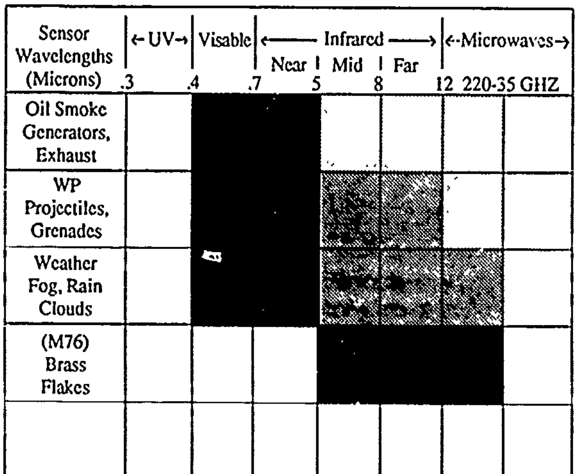


Figure 3. Various obscurants affect different sensor portions of the electromagnetic spectrum.

Methods to Deploy Smoke

The methods used to deploy smoke to the battlefield include grenades, artillery, rockets, fires, and vehicle exhaust. Smoke grenades and artillery have both ground and air burst capability. Armored vehicles such as the U.S. M1-A1 and German Leopard II tanks have smoke grenades that can be launched to obscure their positions, as shown in Figure 4. In this example, a pattern of four smoke grenades will airburst in front of the tank providing complete visual obscurance in less than 0.3 seconds. [6] The resulting smoke screen will persist for several minutes depending on wind conditions.

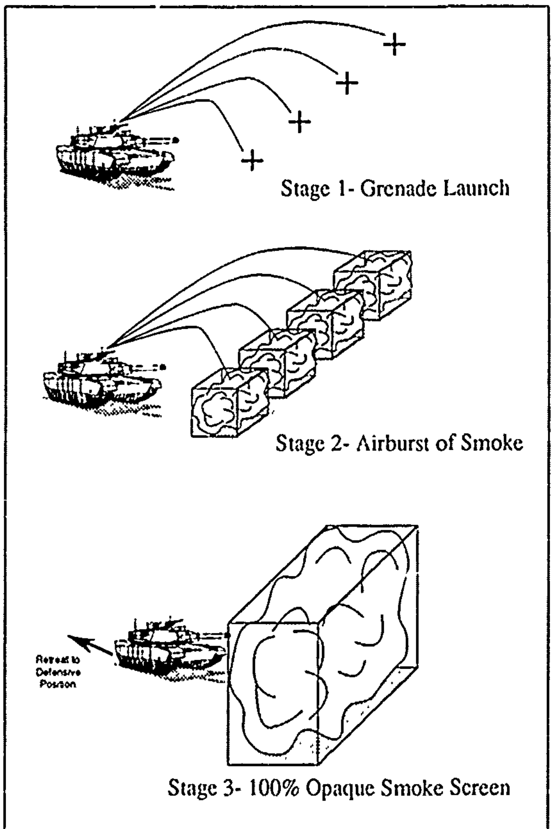


Figure 4 - Smoke grenades are launched from tanks to create a defense smoke screen

Smoke from fires and smoke pots are used to obscure areas of the battlefield. This technique was frequently used in the Gulf Conflict "Desert Storm". [7] During the middle of February 1991, frequent reports by the Associated Press stated that use of smoke from fires hampered military operations.

Smoke artillery is used to visually obscure large areas. The individual artillery shells are aimed in a staggered pattern to fill an entire area with smoke. Additional smoke artillery rounds are used to prolong the effect, or to intensify the reduced visibility to defeat enemy operations.

Tactical Maneuvers Used With Smoke

The principal tactic employed with battlefield smoke is one of short-term self-defense of friendly forces. These tactics have one thing in common: the decision to commit smoke to the battlefield is made rapidly and under pressure.

The self-defense tactics include:
- cover and concealment when unexpectedly threatened by the enemy
- retreat from a fire-fight against a superior enemy
- concealment of a position change from enemy observation
- recovery of failed vehicles

Tactical smoke can also be used as an offensive tool. These tactics are more calculated and are used in coordination with other forces in the area.

The offensive tactics include:
- smoke laid directly on enemy positions to confuse and disorient
- smoke laid behind enemy forces to silhouette for better detection
- decoy smoke launched to confuse or mislead enemy forces
- smoke is used to visual degradation of the battlefield to take advantage of superior sensors.

Visual Simulation
Technical Challenges

This section of the paper will discuss the technical challenges of implementing battlefield smoke within a networked simulation environment. In particular, we will focus on the visual simulation and effects of volumetric smoke.

The physical characteristics of smoke which influenced our determination of an implementation of a smoke rendering technique include the spatial volumetric nature, the variability of visual transmittance, and the dynamic temporal effects. These characteristics of battlefield smoke which impact the visual simulation are well documented by empirical data collected in battlefield smoke studies [4, 8].

Challenge #1
Spatial Volumetric Nature of Smoke

Tactical smoke clouds cover only a portion of the battlefield. Soldiers can be outside the smoke looking in, inside the smoke looking out, inside smoke looking within, or visually unaffected. This characteristic must be preserved from the vantage point of armor troops on the ground and aviation troops in the air, as shown in Figure 5.

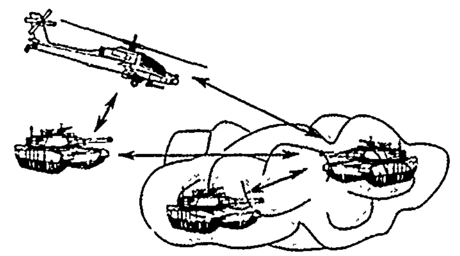


Figure 5 - Smoke is a 3-D volume that must be presented correctly from all vantage points.

Smoke can consist of a single large cloud with a totally homogeneous density, or many individual, localized smoke clouds each with it's own silhouette or geometric shape. The separation of multiple smoke clouds becomes an issue spatially and with respect to viewing depth from a viewpoint. The example in Figure 4 showed multiple smoke clouds from grenades. The visual transmittance associated with a tank sitting between two separated smoke clouds will differ from a tank sitting within two overlapping smoke clouds.

The key factors are the shape of the smoke cloud and the transmittance function through the cloud. Smoke requires more than just an exterior geometric definition. It must be thought of as a volume with its interior being as important as its geometry. This information is necessary for determining the visible effect of the smoke volume(s) on objects on the battlefield. This interior characteristic of the smoke cloud is discussed in the next section on transmittance.

Challenge #2
Variable Transmittance of Smoke

Transmittance is a quantitative measure of the ability to see an object within a smoke cloud A value of 1 means clear viewing and no effect from smoke; a value of 0 means total smoke opaqueness. The attenuation, or visible change to an object's color due to the smoke cloud, is a function of the smoke density and the depth of the object into the smoke along the ray from the viewpoint, as shown in Figure 6.

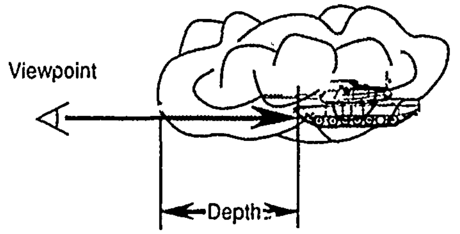


Figure 6 - Transmittance is based on the objects depth into the smoke and the smoke density.

The following equations describe attenuation through the atmosphere.

Equation 1: Attenuation = (1 - W)

where: W is the transmittance weight

If the cloud density is assumed homogeneous for a sampled ray through the volume, then radiant energy transport theory suggests a Poisson attenuation function for points within the cloud.

Equation 1.1: $W = \exp(-\beta z)$

where:
z is the depth into the smoke cloud
β is a factor for types of smoke

The attenuation weight values for different types of smoke can be estimated using various values of β in Equation 1.1, as shown in Figure 7.

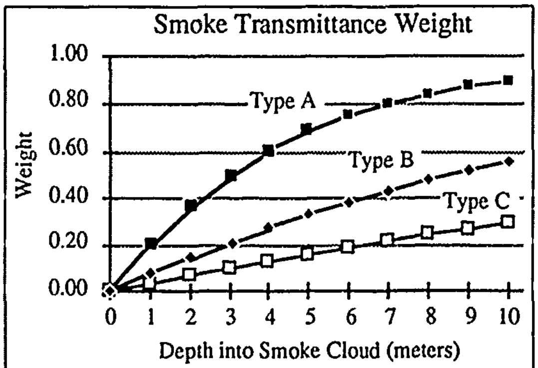


Figure 7 - Transmittance varies based on the type of smoke cloud.

Sensor systems acquire various spectral bands as discussed in the first part of this paper. Different smoke types have different spectral characteristics, providing varying transmittance as viewed through a sensor. For example, red phosphorous smoke works well to defeat visual and near-IR spectrum sensors and M76 brass flakes defeat far-IR and millimeter wave regions of the spectrum. [4]

The overall variability of transmittance profiles for various sensors and visual modes must be modeled for a visual smoke capability in networked simulation.

Challenge #3
Temporal and Dynamic Smoke Features

The physical characteristics of smoke shape, size and density are dynamic. Physical changes are largely due to changing atmospheric conditions such as windspeed and temperature. For example, a smoke grenade after launch, emission, and dissipation, can move two kilometers down wind, and grow to 600 meters in width within 7 minutes. [4]

The density of various smoke types can change extensively over time. For example, the density of brass flakes (M76) will vary from 5 to 0 grams/m3 in less than 20 seconds. Under the same atmospheric conditions, red phosphorous smoke will vary from 2.5 to 0.3 grams/m^3 in 5 minutes. [4]

CIG Technology Limitations

Current Computer Image Generation (CIG) technology has several fundamental limitations with respect to rendering localized volumetric battlefield smoke effects. These limitations include areas such as planar geometric modeling versus volumetric, restrictive occulting methods, homogeneous transmittance models, and graphics performance limitations.

Limitation #1 - Geometric Modeling

The current generation of real-time CIG systems model all geometry with planar or polygonal primitives. This is appropriate for solid objects and thin translucent surfaces such as windscreens, but does not appropriately define a volume with internal attributes such as density.

Early experiments with smoke visual simulation used simple 3-D geometric shells. This approach provided the outer surface definition of a smoke cloud, although no internal density profile was maintained.

Limitation #2 - Occulting Methods

The occulting requirements for smoke include the need to handle large numbers of dynamic objects in the scene, and to handle partial occulting of semi-transparent objects.

The current generation of real-time visual systems is based on one of two hidden surface elimination methods or a hybrid of the two. These methods are depth buffer, and binary space partition (BSP) techniques.

Both techniques work well for performing basic occulting or sorting of planar polygonal surfaces. The depth buffer is better suited for handling multiple dynamic objects, as it sorts objects implicitly at the pixel level. BSP systems are efficient with only relatively few dynamic objects. (i.e. less than a dozen)

These two methods are not naturally amenable to handle localized volumetric densities of smoke. This means that simple front-back relationships are not good enough to determine partial occulting for semi-transparent objects in the scene. Therefore new supplemental or modified methods are required to handle hybrid primitive collections of planar and volumetric methods.

Limitation #3 - General Transmittance

Current CIG technology typically incorporates a homogeneous atmospheric effect technique that globally affects all things in the scene. This method relies on a predefined function based on the distance from the viewer into the scene (depth) to determine the appropriate level of haze attenuation to apply to objects. This method will not handle multiple localized smoke volumes that cover different spatial sections of the screen.

Limitation #4 - Graphics Performance

CIG system performance is measured by both polygon throughput and pixel throughput or depth complexity. [9] Polygon performance requirements are a function of database complexity including parameters such as terrain complexity, density and complexity of static cultural features, and vegetation models. In our networked simulation applications, we have experienced substantial additional polygonal loading because of moving objects and special effects such as bomb bursts, dust, and smoke.

Pixel processing requirements in networked simulation are largely driven by three parameters, database roughness, object density, and dynamic object and special effect occurrences. The first two can be well planned in the database engineering of an environment, but special effects are typically uncontrolled. For example, a tank commander may launch a dozen smoke grenades resulting in very high pixel requirements.

Visual Simulation Technical Innovations

This section discusses the innovations for the visual simulation of smoke. A volumetric rendering method is presented which has been implemented in real-time CIG hardware for networked simulation and training systems. [10]

Innovation #1 - Volumetric Definition

To properly manage multiple localized overlapping volumetric smoke processing within a hybrid depth buffer architecture, we developed a technique which manages multiple smoke volume buffers in addition to the standard color and depth frame buffer elements, as shown in the Figure 8. [11]

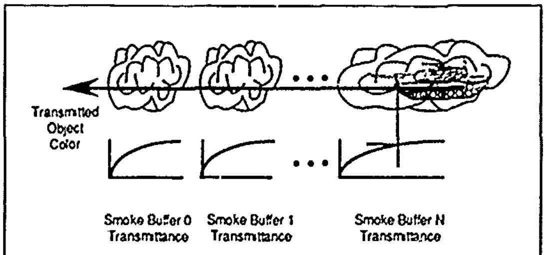


Figure 8 - Multiple smoke volumes simulated.

The smoke volume buffers must be separately managed to allow proper attenuation of non-smoke volume objects such as targets that fall within or beyond smoke volumes.

In this way a formula for managing multiple smoke volumetric attenuation levels is adopted:

Equation 2: $$W_i = w_i \left[\prod_{k=1:n,\ k\neq i} (1-w_k)\right] (1-w_h)$$
for all k in front of i, $(Z_i - Z_k) > 0$

where:
W_i is the attenuated transmittance for cloud i.
w_i is the transmittance level of ith cloud as f(z)
$(1-w_k)$ is the attenuation of the kth cloud
$(1-w_h)$ is the haze attenuation at the ith cloud
Z_i is the depth to front of cloud i
Z_k is the depth to front of cloud k

The non-smoke transmittance weight contribution (W_o) is as follows:

Equation 3: $W_o = [\Pi(1-w_i)]\ (1-w_h)$
$i=1{:}n$, for all i in front of p,
$(Z_i - Z_o) > 0$

where:
W_o is the object transmittance weight
Z_o is the depth to front of object surface

Equation 4: $W_h = 1 - \Sigma(W_i \times W_o)$

where:
Wh is the attenuated transmittance for haze

These formulas can then be used to sum each element of transmitted color times it's weight (W_i) level:

Equation 5: $C_p = [\ \Sigma(W_i * C_i)\ ] + (W_o * C_o) + (W_h * C_h)$
$i=1{:}n$, for all i

where:
C_p is the transmitted color
C_i is the color of cloud i
C_o is the color of object in the scene
C_h is the color of haze

The result represents the appropriate color occurrence at a given pixel location. Additional opaque and transparent pixels can subsequently have smoke applied and potentially be combined into the same pixel location.

Innovation #2 - Hybrid Occulting Method

Since we are focused on networked training systems with hundreds of moving vehicles, we investigated a modified hybrid depth buffer approach for integrating volumetric and planar hidden surface removal and rendering.

The volumetric smoke rendering method presented here is integrated into a hybrid depth buffer CIG rendering system This requires that depth and density elements of multiple smoke buffers per frame buffer pixel be retained to allow proper smoke application. Smoke buffer pixels cannot be processed and thrown away since the depth and density values are necessary to properly calculate the attenuation and transmittance of each smoke cloud.

The smoke buffer attenuation is then applied with a "depth buffer" method modifying Equations 2.0 thru 5.0. The depth buffer tests, (Zi - Zk) > 0 and (Zi - Zo) > 0 are used to determine whether a smoke buffer element is nearer than the non-smoke entity and if attenuation should be applied. In the same way w_i is modified through a depth difference ($Z_i - Z_o$) to determine if a non-smoke entity (o) is within the smoke cloud i. This will allow smoke weight w_i to smoothly transition to zero opacity as an object moves through the smoke cloud.

Innovation #3 - Variable Transmittance

The transmittance model for smoke must be variable for different types of smoke as well as variations in spectral sensing. [12] The w_i in Equations 2 and 3 above must follow a modeled transmittance profile such as that shown in Figure 7. The depth at which an object falls within the smoke volume must then use the proper transmittance along the curve to attenuate the object transmittance level. Various transmittance profiles are stored to allow for varying density levels associated with wind, temperature, and other environmental conditions. A parametric transmittance table can allow selection of the proper transmittance profile based on meteorological conditions. Additionally the table allows for variations based on the sensed spectral band or visual spectrum.

Innovation #4 - Efficient Processing

We determined it was too costly to model complex smoke shapes with the traditional method of many front and back facing planar facets to define a volumetric region. We came up with an efficient method of using a single gimbaled polygon, which has a complex depth offset representation defined within the applied texture map as shown in Figure 9. The transmittance attributes are also stored within the texture map. This turns out to be very useful in reducing the polygonal load while still allowing complex geometric smoke shapes.

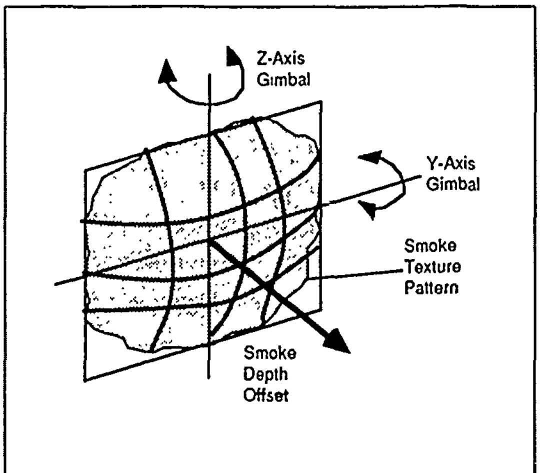


Figure 9 - Gimbaled smoke with depth offsets.

Summary and Future Work

This presented approach is a unique visual simulation of volumetric atmospheric obscurance. The resulting visual representations of tactical smoke are very realistic as shown in Figures 10 & 11. BBN is continuing refinement of the algorithms and applications for tactical team training.

Future work has been identified to investigate the smoke impacts to networked simulation such as wind effect simulation, Semi-Automated-Forces (SAF), and intervisibility determination.

Figure 10 - Computer generated image of tactical smoke clouds in a village.

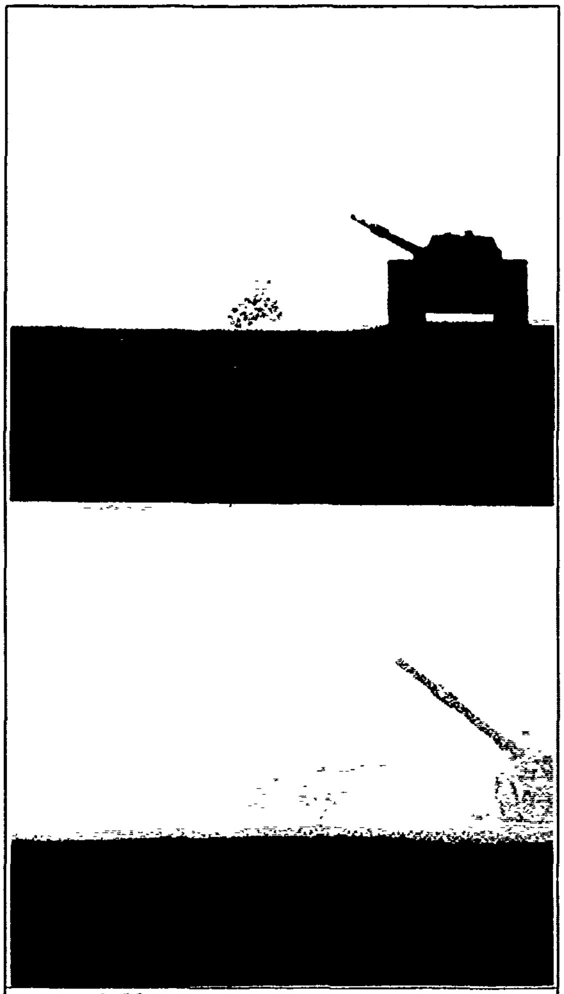


Figure 11 - Computer generated image of smoke in tactical training situations.

References

[1] LtCol (P) Jack A. Thorpe, USAF (Ph.D) "Warfighting with SIMNET - A Report from the Front," Proceedings from I/ITSC, 1988

[2] Independent Verification and Validation of the SIMNET Model, Potomac Systems Engineering, Inc. 5 Sept 1990

[3] Written interview, "Tactical usage of battlefield smoke", Wegmann April, 1991

[4] Proposed Architecture to Include WP Smoke Within SIMNET, GEOMET Technologies, Inc. 9 May 1989

[5] Developmental SIMNET Data Handbook, DARPA July 31, 1986.

[6] 76 mm smoke grenade launcher specification, Wegmann and Buck, West Germany

[7] Dilip Ganguly, The Associated Press, Seattle Post-Intellegencer, Feb 14, 18, 1991

[8] Draft Report 4.2 Inch Smoke Study, Vol I Technical Solutions, Inc Sept 1987

[9] Rick Bess, BBN Systems and Technologies "Tradeoffs in the Configuration of Computer Image Generation Systems," Proceedings from I/ITSC, 1989

[10] Patent Application in process.

[11] Smoke Rendering Equation for Multicloud, Multisurface Model. J Ferguson, B. Soderberg, BBN ASD Graphics Design Working Note 44.

[12] Evaluation of Atmospheric Effects for Operational Tactical Decision Aid, Penn State University, Universal Energy Systems, Aug 1986

ABOUT THE AUTHORS

Rick Bess is currently the Product Manager for Visual System Products in the Systems and Technologies Division of Bolt Beranek and Newman. Mr. Bess has a long history with networked visual systems, having worked at the Boeing Advanced Computer Graphics Technology Lab in 1982, and having been a 1984 core team member of the SIMNET distributed simulation and training program. His related technical background include being a senior software engineer developing CIG database creation systems, and graphics algorithm design. He has a Bachelor of Science degree in Aerospace Engineering from the University of Kansas.

Brian Soderberg is currently the Project Engineer for BBN's New Visual System Product Development Programs. Mr. Soderberg has been active in numerous research teams to study the simulation of Battlefield Smoke. Previously, Brian held the position of senior software engineer involved in graphics algorithm development. Since 1982, Brian has been involved in computer graphics research and development with techniques involving depth buffers, lighting models, texture mapping, surface rendering, motion design, and ray tracing. In addition, Brian has lectured on computer graphics at various companies and at the University of Washington. Brian has a Bachelor of Science degree in Mechanical Engineering from the University of Washington.

ANTIALIASING WITHOUT SUPERSAMPLING

Walter Gish
Allen Tanner
Terabit Computer Engineering
Salt Lake City, Utah

ABSTRACT

Computer generated visual images can exhibit a variety of artifacts known collectively as aliasing. These artifacts are distracting and counterproductive to the training process. Thus removing these artifacts through antialiasing has become an important characteristic of every modern image generator.

While a variety of antialiasing techniques exists, image generator manufacturers have settled on a technique called supersampling. Supersampling computes very high resolution images and filters them down to match the resolution of the display. The cost of computing these higher resolution images is significant.

This paper introduces a new approach to antialiasing that operates at the displayed resolution, without resorting to the added burden of generating a higher resolution image. The ramifications for an emerging generation of compact, low cost display generators and visual simulators are discussed.

INTRODUCTION

Antialiasing is widely recognized as a required feature of high quality display and image generators.* Without antialiasing digital images exhibit a variety of distracting artifacts:

- jagged or stairstepped edges (which move back and forth like escalators as edges move)
- crawling of edges
- breakup of thin polygons
- scintillation of small objects

Some of these artifacts (jagged edges and the breakup of thin polygons) are static - they occur even for still images. The others (escalators on edges, crawling edges, and scintillation) are dynamic - they occur only with moving images. These artifacts are distracting and counterproductive to the training process. [20]

This paper examines two aspects of antialiasing:

- the quality of the antialiased image
- computational complexity of antialiasing

Image quality is measured by the degree to which aliasing artifacts are reduced and the absence of ringing and excessive blurring. The computational complexity of antialiasing controls implementation complexity and ultimately system cost.

The most common method for antialiasing polygons is supersampling. In supersampling a higher resolution image is generated and then filtered down to the resolution of the display. This higher resolution image has 4 to 8 times the resolution of the display in both the x and y-directions. [20] Computing the higher resolution image is equivalent to recomputing an image at the displayed resolution many times over. Thus the total cost of computing and filtering the higher resolution image can be enormous. While this cost may be reduced by decreasing the degree to which the higher resolution image exceeds the displayed resolution, image quality is sacrificed - either aliasing artifacts remain or the image is blurred.

Although supersampling is commonplace for antialiasing polygons, its adequacy is questionable for small details, such as wires or thin polygons. For example, as a polygon is rotated away from the viewpoint it appears thinner and thinner until it becomes, in essence, a line**. For lines supersampling is useless. Antialiasing lines (and characters) is difficult and requires special techniques. The best known technique for antialiasing lines uses a one-dimensional approximation (the distance from pixel centers to line center) of the geometry to derive the filter result. [12] Thus the technique most commonly used to antialias lines is very different from the technique employed for polygons.

Because the cost of generating and filtering a higher resolution image for supersampling is significant, we propose abandoning supersampling altogether. Instead the system described here directly filters the shape of a polygon or line over an area surrounding each pixel. In terms of image quality, this reduces aliasing artifacts beyond the limits of visual perception. More importantly, it is no longer necessary to compute a higher resolution image. In turn this allows the implementation to be much more cost effective. The result is a clear, sharp, artifact-free image at low cost.

The next section reviews the basic principles that explain how aliasing arises and what can be done to remedy it. The following sections then describe a new approach to antialiasing we call shape filtering, its implementation, and its ramifications.

* The term display generator means a system which displays lines and polygons with at most Gouraud shading and a painter's algorithm for occulting. Whereas, an image generator has full 3D hidden surface processing, texture mapping, and more sophisticated shading.
** The term line means an infinitely thin, mathematical line.

ANTIALIASING TECHNIQUES AND THEIR EFFECTIVENESS

Aliasing in digital images arises from the 2-dimensional sampling process. With computer generated images a mathematical idealization of the intended image is processed to produce pixels. This mathematical ideal considers the intended image to be a continuous function of intensity (for black and white images) or color over a 2-dimensional (i.e., x-y) domain. Sampling this continuous image onto a discrete array of points causes high frequency components to be aliased onto lower frequencies. The prerequisites for sampling to provide an accurate representation of the intended image are given by a two-dimensional version of the well known sampling theorem. The sampling theorem states that if a signal has no energy above half the sampling frequency, the original signal can be exactly reconstructed from its samples. This cutoff frequency, half the sampling frequency, is known as the Nyquist frequency.

The amount of aliasing introduced is dependent upon the frequency spectrum of the continuous image being sampled. Now computer generated images are composed of line and polygon primitives. Therefore an examination of the Fourier spectra of a line and a polygon edge provides insight into the aliasing process. Consider an infinite edge aligned along the y-axis and a line also lying on the y-axis as shown in Figure 1(a). We can simplify the mathematics by considering only the component of the Fourier spectrum in the direction perpendicular to the edge or line, i.e., the x-direction. In this direction the edge and line have intensity functions shown in Figure 1(b). These are the step function and delta function. In the limit, these have Fourier transforms [4]

$$F_{step}(f) = \frac{1}{\tau f} \qquad F_{delta}(f) = 1$$

which are sketched in Figure 1(c). Note that both edges and lines have Fourier components out to infinite frequency. No matter how finely edges and lines are sampled, the result is always aliased. Also, because the frequency content of the ideal line never falls off, sampling is useless for rendering lines. Having a sample lie on an arbitrary line is a hit-and-miss proposition. Therefore lines require special rendering techniques both for the aliased case (Bresenham's algorithm [6]) and for the antialiased case (Gupta-Sproull algorithm [12]).

To reduce aliasing it is necessary to reduce the frequency content above the Nyquist limit through filtering. Suppose we chose a filter k(x) whose integral is K(x) as shown in Figure 1(d). The region over which this filter is non-zero is called the filter domain. After filtering, the step function of the edge becomes K(x) and the delta function of the line becomes k(x) as shown in Figure 1(e). Figure 1(e) shows that any antialias filtering blurs or "widens" the original edge and line. The selection of filters for antialiasing involves image quality trade offs between aliasing, blurring, and ringing. [3] [17] For example, to remove all energy above the Nyquist frequency requires a sinc filter, [3] but this introduces

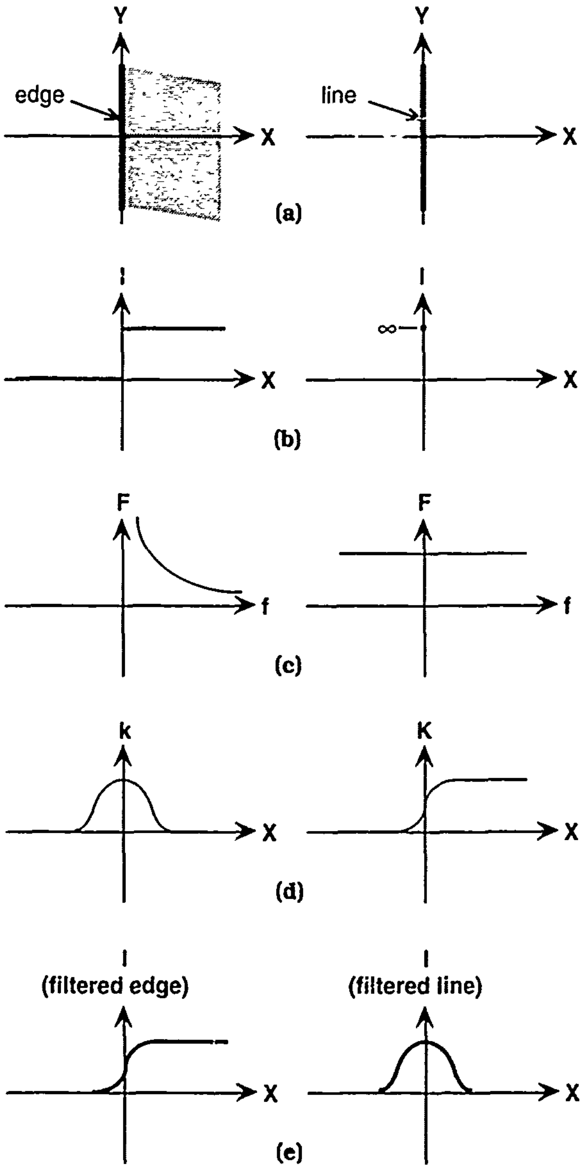


Figure 1. Spectra of Ideal and Antialiased Lines and Polygons.

ringing artifacts. So in practice, antialias filtering never removes all the energy above the Nyquist frequency and there is always some residual aliasing. The goal is to make the residual aliasing imperceptible without degrading image quality.

While the choice of filter is important, how the filtering is performed is even more important. There are two basic ways to perform this filtering. Ideally, the filtered intensity I at a pixel located at x is given by the continuous integral (often called a convolution integral):

$$I = \iint_{domain} k(u)\, i(\bar{x}-\bar{u})\, d\bar{u}$$

where k(x) is the filter and i(x) is the original (unfiltered) image. This is called prefiltering because the filtering is performed prior to sampling at pixel resolution. The other way to perform this filtering is to approximate the

continuous integral by a discrete sum over samples of the continuous image:

$$I = \sum_{n \in \text{domain}} k_n \times i_n$$

This is called supersampling because many samples of the continuous image are required for each output pixel Supersampling is universally used for antialiasing polygons in hardware and also used by some software packages (such as Renderman™). The problem is that supersampling creates the following issue. how many samples are necessary to adequately approximate the filtering? More samples give a better approximation (and a better image) but also require more computation.

To reduce aliasing artifacts to an imperceptible level for still images requires between 16 samples over a 4-by-4 sub-pixel grid and 64 samples on an 8-by-8 grid (for regularly spaced samples). [9] Motion places even further demands (i.e., more samples) on antialiasing. Computing so many samples affects system performance and cost. Consider an antialiased z-buffered image generator. For the case of a single processor per pixel, the real-time performance is degraded proportionately to the number of samples required. [13] Conversely, when multiple hardware units for each pixel are used to achieve real-time performance, there is a proportionate increase in cost. The availability of custom VLSI has made such image generators possible, but not inexpensive.

Given this situation it is not surprising that several efforts have sought to reduce the number of samples required or to simplify the problem. One approach has been to use sub-pixel bit masks during scan conversion. [7] [20] But this requires either low resolution masks or constant (area) filters. The need for low resolution with arbitrary filters can be understood from the following argument. A 4-by-4 sub-pixel mask requires a 2^{16} = 65,536 word filter table, but doubling the resolution in just one dimension to a 4-by-8 mask (as in the A-buffer [7]) requires a 2^{32} = 4,394,967,296 word table! Thus, arbitrary filters are possible only at low sampling densities. Alternatively, for the constant or area filter the result can be obtained by simply counting the number of ones in the bit mask.

Another approach, which also can be used with sub-pixel bit masks, is to space the samples non-uniformly. [8] Non-uniform sampling effectively reduces the regularity of jagged edges, the most obvious aliasing artifact, with fewer samples than uniform sampling. However, non-uniform sampling does not overcome the fundamental limitations inherent with supersampling. The basic problem with supersampling is that it discards all the information in the image except at the infinitely small sample points. Using fewer samples means that even more information is discarded. Therefore, any reduction in the number of samples aggravates supersampling's fundamental limitation. As computer generated images become increasingly realistic (i.e. more polygons, etc.), the information discarded is more and more significant. This is why so many samples are required. It takes many samples to be sure something important hasn't been missed. Even the most sophisticated techniques require 40 samples per pixel just for still images. [14] This problem with supersampling is well known to experts. For example, Crow recently wrote that the possibility of missing important detail between samples is the "principal objection to supersampling as a remedy for antialiasing." [10]

Cost constraints force today's image generators to calculate typically 8 to 16 samples per pixel. With a filter defined over a 2-by-2 pixel area (such as the pyramidal filter shown in Figure 6) this results in 32 to 64 samples contributing to each displayed pixel. While this has the appearance of adequacy, it is the sampling density that is important. Occasionally aliasing artifacts remain, such as scintillation or the break-up of thin polygons. Also, using few samples often requires choosing a filter that excessively blurs the image in order to reduce dynamic aliasing artifacts. Figure 2 shows an example of this blurring. Here high quality antialised lines (a raster HUD) have been optically overlaid onto an image from a real-time image generator. If the image generator antialiased with the same fidelity as the line generator, then the edges of the aircraft or the mountain range would be just as sharp as the lines. However, Figure 2 shows that they are not

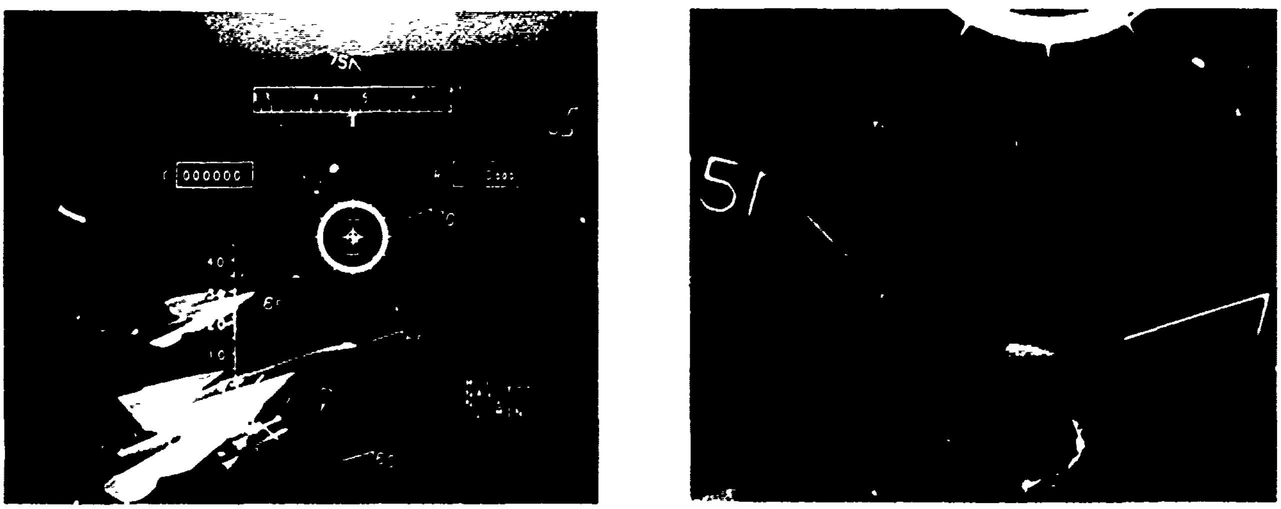


Figure 2. A typical real-time image with overlaid antialiased raster lines.

(this is more visible in the enlargement at the right).

We leave it for others to consider the need for higher image quality in real-time image generators. What we can say here is that there is an alternative antialiasing method that addresses the shortcomings of supersampling.

SHAPE FILTERING

The flat frequency spectra of lines makes them harder to antialias than polygons. As mentioned previously, lines are almost universally antialiased by approximating the geometry of line and pixel by a single parameter - the distance from the line to the center of a pixel. This distance is then used to look up a precomputed filter convolution in a table. [12]

Suppose now that we take a similar approach for polygons. That is, instead of approximating the continuous filter integral by a sum over many samples, we approximate it by the continuous integral over a "shape" very close to the shape of the original line or polygon. We call this **shape filtering**. In particular, the approximate shape is obtained by simply rounding any vertices within the filter domain to the nearest discrete value on some high resolution grid. Figure 3 shows this rounding for a high resolution grid of 1/8th of the pixel spacing. Note that rounding simply moves vertex A up and to the left, vertex B down and to the right, and vertex C up and to the right.

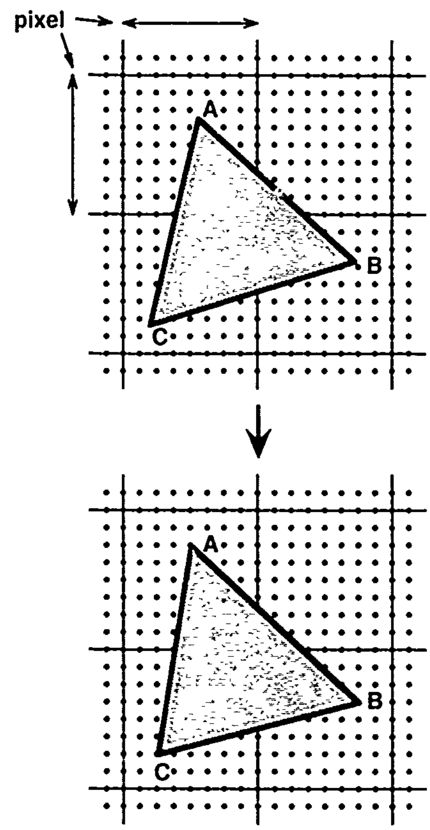


Figure 3. Rounding vertices for shape filtering.

This is fundamentally different from supersampling. With supersampling the error comes from approximating the continuous integral by a finite summation. In contrast, with shape filtering, there is only "shape approximation" error. Therefore the factor determining image fidelity changes from how many samples are needed (as in supersampling) to how accurately must shape be represented?

Since the shape approximation is just a perturbation of vertices, shape accuracy is proportional to the precision with which vertices are represented. How much precision is necessary? For black and white images greyscale (which is equivalent to antialiasing in this context [15]) can be used to infer the position of edges to sub pixel accuracy, possibly even as much as 1/16 of the pixel spacing (4 sub-pixel bits). [16] Several recently introduced graphics workstations use a value of 1/8 of the pixel spacing (3 sub pixel bits) for line antialiasing calculations and to ensure smooth motion. [2] For edges the rigorous answer comes from a recent experiment [18] in human visual perception. This experiment directly measured the effective sub pixel positioning that can be inferred using greyscale. The result was that no more than 3 bits of sub pixel location accuracy was perceivable. Using more bits doesn't improve anything. Thus our system adopts a 1/8 pixel resolution because this is the limit of human visual perception.

How does shape filtering at 1/8 pixel resolution remedy aliasing artifacts? Let's reconsider the four kinds of aliasing artifacts previously noted:

- Edge stairstepping
- Edge crawling
- Breakup of thin polygons
- Scintillation of small objects

Edge stairstepping and crawling can be reduced beyond the limits of human perception because of the edge positioning cited above. For the breakup of thin polygons, recall that shape filtering handles lines and thin polygons consistently. If very thin polygons (less than a fraction of a pixel) are displayed as lines with appropriate intensity these artifacts are also eliminated. The scintillation of small objects can be eliminated by clamping the size of small objects to some consistent value before rendering.

Even when thin polygons are not transformed into lines and small objects are not clamped to some size, shape filtering reduces these artifacts substantially. This is because shape filtering is more accurate than supersampling on an 8 by 8 sub pixel grid. A quantitative comparison of supersampling and shape filtering is both complex and dependent on the particular filter. Still a rough comparison for the constant or area filter is easy. In this case supersampling on an 8-by-8 grid determines area in increments of 1/64 of the pixel area (i.e., 0/64, 1/64, ... 64/64). In contrast shape filtering determines area in increments of 1/128 of the pixel area. While worst case error is 1/8 of the pixel area for

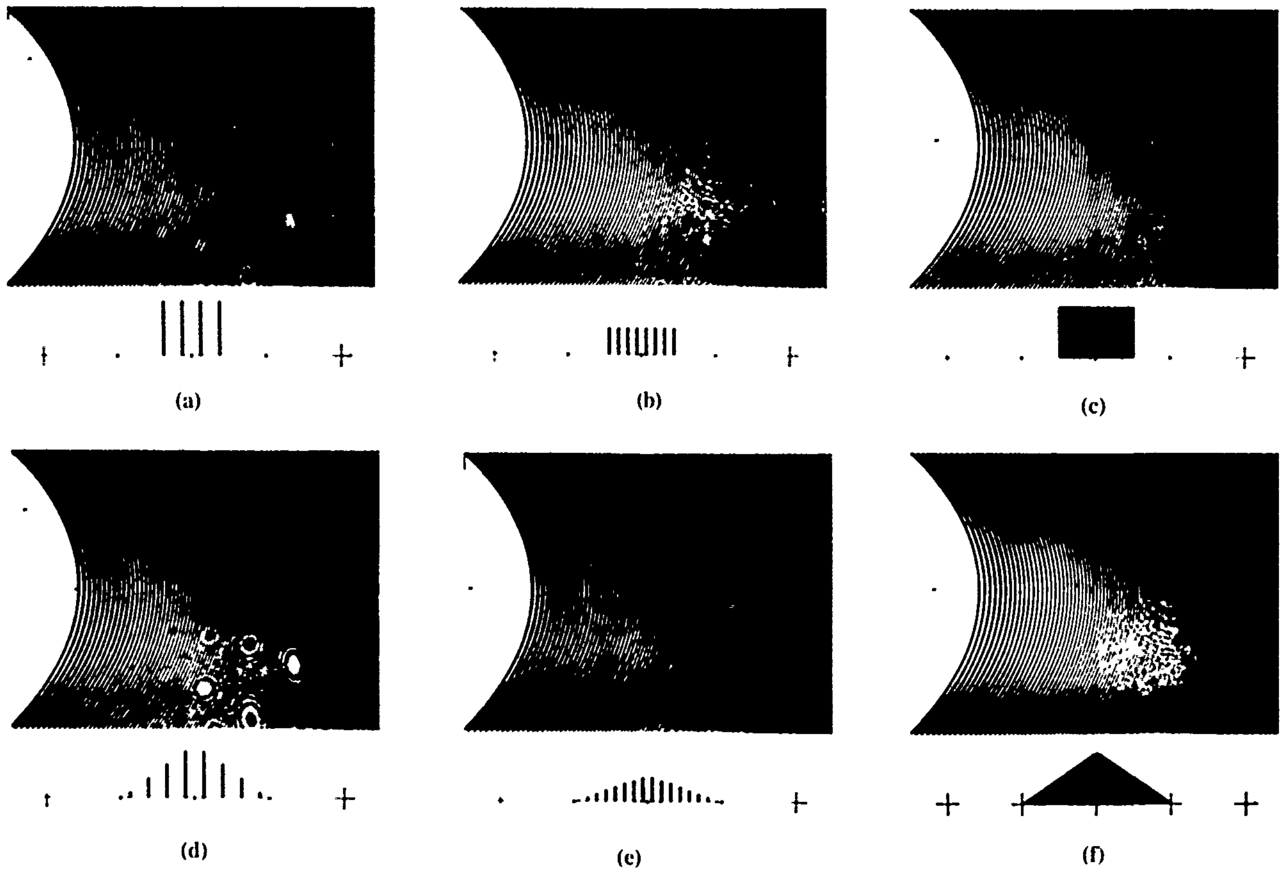


Figure 4. Antialiasing test patterns.

either approach, there are many cases where shape filtering is more accurate.

Finally, we can visually compare supersampling and shape filtering. Figure 4 shows a test pattern used to compare antialiasing effectiveness. [10] Figure 4 compares two filters

- uniform (constant) filter (Figure 4a, 4b, 4c)
- triangular (pyramidal) filter (Figure 4d, 4e, 4f)

using three different implementations of the filtering process:

- 4-by-4 or 16 samples per pixel (Figure 4a, 4d)
- 8-by-8 or 64 samples per pixel (Figure 4b, 4e)
- continuous integral (Figure 4c, 4f)

Two observations are in order. First, the quality differences between the two filters are less significant than the differences between the three implementations. At the lowest sample density (4-by-4 sampling), variations between different filters are dwarfed by the inaccuracy that 16 samples approximate the correct filter result. Secondly, at a density of 8 by 8 or 64 samples per pixel, image quality is approaching that of the continuous integral (Figure 4b and 4c or 4e and 4f look similar).

In summary, shape filtering approximates shape to the nearest 1/8th of a pixel in both x and y directions, but within this constraint, the filtering is exact. By maintaining position error to 1/16th of a pixel (rounding to the nearest 1/8th pixel) the antialiasing fidelity is commensurate with the limits of human vision. Shape filtering is better than sampling at a density of 64 samples per input pixel - about an order of magnitude more samples than existing real-time systems presently calculate.

HARDWARE IMPLEMENTATION

This section describes an implementation of shape filtering built using only off-the-shelf TTL and CMOS parts. While the following description is necessarily an overview of the architecture, details are available elsewhere. [11]

The system scan converts, antialiases and Gouraud shades lines and polygons. The filters are arbitrary over a 3-by-3 pixel domain for lines and a 2-by-2 pixel domain for polygons shown in Figure 5.

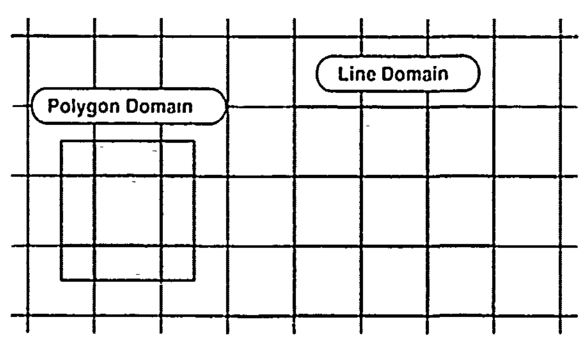


Figure 5. Filter domains for shape filtering.

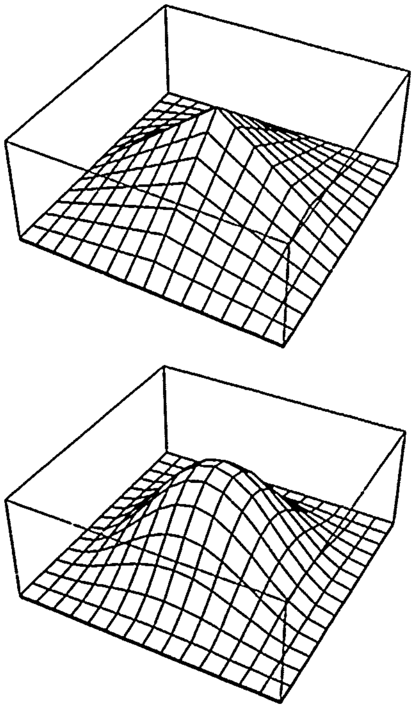


Figure 6. Two-dimensional pyramidal and cos^2 filters.

The 2-by-2 pixel domain for polygons allows filters such as the pyramidal and cos^2 filter shown in Figure 6, as well as the uniform or constant filter. The larger 3-by-3 domain for lines allows the use of filters that not only antialias lines, but can give them varying degrees of thickness.

The implementation itself is an enhancement to conventional aliasing scan conversion. [1] Figure 7 shows the functional steps in this process.

The first step, trapezoidal decomposition, breaks down

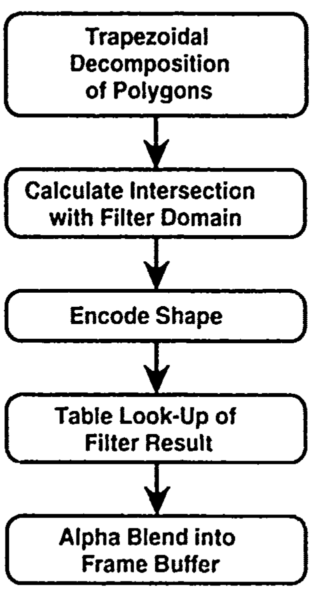


Figure 7. Scan conversion with shape filtering.

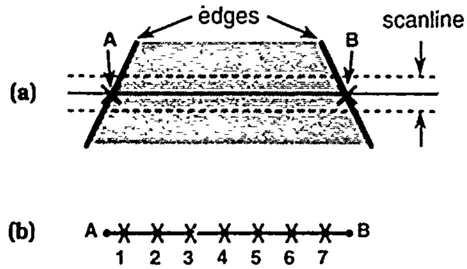


Figure 8. Interpolation for conventional (aliasing) scan conversion.

arbitrarily shaped polygons into a set of horizontally aligned trapezoids. Many systems (e.g., graphics workstations) perform this step only for a restricted set of input polygon shapes (e.g., triangles and quadrilaterals). Our system accepts arbitrary polygons - any number of sides - convex or concave - with or without holes. Decomposition techniques for the general case are well known. [5]

The second step calculates the intersection of the original line or a trapezoid (from the polygon decomposition) with the filter domain for each pixel. This step consists of a series of interpolations along lines and along and between polygon edges. For shape filtering, the interpolations for polygons are nearly identical with those performed by conventional aliasing scan conversion (as performed in graphics workstations). [1] In aliasing scan conversion first the x-coordinate and the color are interpolated at the center of each scanline (points A and B in Figure 8(a)). Then color is interpolated at pixel centers from the x-coordinate of the left edge to the x-coordinate of the right edge (points 1 through 7 in Figure 8(b)):

Thus conventional (aliasing) scan conversion consists of interpolation in the y-direction of an edge's x-coordinate and color, followed by interpolation in the x-direction of color.

Shape filtering performs similar interpolations. As with aliased scan conversion, there is first an interpolation in the y-direction of an edge's x-coordinate and color. The difference is that now these are calculated at the top and bottom of the filter domains for each scanline as shown in Figure 9(a). Since, the bottom of one filter

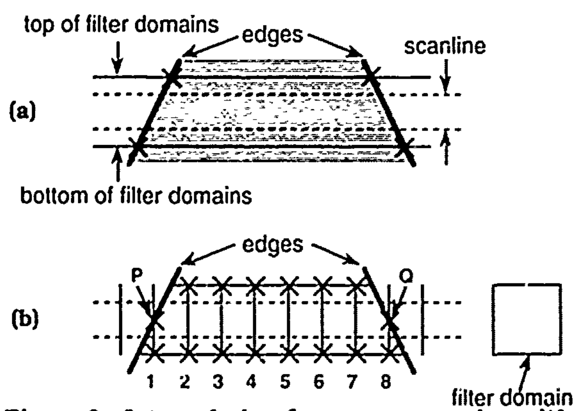


Figure 9. Interpolation for scan conversion with shape filtering.

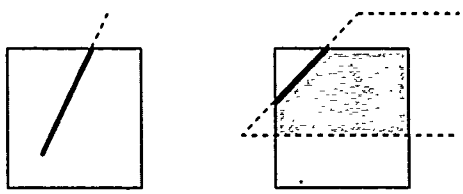


Figure 10. Typical line and polygon shape within a filter domain.

domain is also the top of another domain for a nearby scanline, these interpolated values are always shared between two scanlines. Thus the number of interpolations required in the y-direction is the same (to first order) as for conventional scan conversion. After the y-direction interpolations, color is interpolated in the x-direction at the boundary between filter domains (points 1 through 8 in Figure 9(b)). Shape filtering also requires calculating the intersection of edges (or a line) with the imaginary vertical lines separating the filter domains horizontally as shown by points P and Q in Figure 9(b). These intersections require an accuracy of only 5 bits (to cover the 3-by-3 domain at 1/8 pixel accuracy), and therefore their calculation requires little circuitry.

The calculated intersections are rounded to the same 1/8th pixel grid that the original vertices or line endpoints are represented. The appropriate combination of these intersections is assembled to yield the shape within the filter domain surrounding each pixel.

For lines, this "shape" is simply an arbitrary line segment within the filter domain. Hence for lines, shape filtering does not require the one-dimensional approximation described previously. [12] This provides more accurate filtering, particularly at endpoints and where lines meet. [11]

For polygons, the shape of possible intersections with a filter domain is more complex. [11] These shapes consist of one or two simpler shapes we call fragments. A fragment consists of an edge (which defaults to the left boundary of the filter domain) and possibly an extra y-coordinate specifying the top or bottom of the trapezoid. Figure 10 shows examples of typical line and polygon intersections.

If the raw shape for either line intersections or polygon fragments was naively used to address a table of precomputed filter results, the necessary table size would be prohibitively large. To keep the filter tables to a manageable size the system encodes shape. [11] The underlying idea of this encoding is simple. Since any reasonable antialiasing filter possesses horizontal and vertical symmetry, two input shapes differing only by reflection about a symmetry axis give the same filter result. For example, in Figure 11(a) the two lines have the same filter result because of the horizontal symmetry of the filter. Figure 11(b) shows two regions that are equivalent because of the vertical symmetry of the filter.

Every symmetry axis reduces the number of independent cases (i.e., shapes) by almost one half. Line encoding uses horizontal, vertical, and diagonal symmetry to reduce the number of independent cases to 56,875. Polygon encoding uses horizontal and vertical symmetry to reduce the number of independent cases to 69,632. For both lines and polygons, the filter table is stored in a single 1 Mbit RAM chip (128K-by-8).

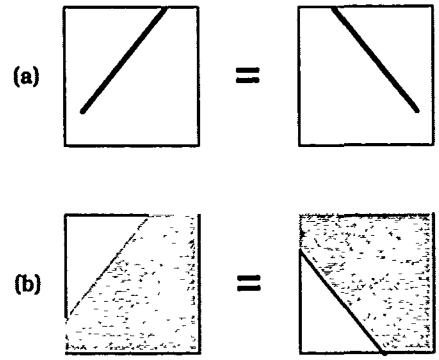


Figure 11. Shapes equivalent due to symmetry.

The filter table itself contains the area integral of the polygon for each tabulated fragment shape:

$$I_p(\text{shape}) = \iint_{\text{shape}} k_p(x,y)\, dx\, dy$$

or the line integral of the line for each tabulated line segment:

$$I_L(\text{line}) = \int_{P_1}^{P_2} k_L(x,y)\, ds$$

The filter weight obtained from the table is multiplied by the opacity (opacity = 1 - transparency) of the polygon or line, and the result is called α. This α controls the blending of the new color with the existing color (or the background color) in the frame buffer. Color can be blended according to the rules of compositing: [19]

$$C_{FB} = C_{FB} + \alpha_{IN} * (C_{IN} - C_{FB})$$

$$\alpha_{FB} = \alpha_{FB} + \alpha_{IN} * (\alpha_{IN} - \alpha_{FB})$$

where C stands for any color component, and the subscripts FB and IN designate the current frame buffer contents and the input values coming into the frame buffer respectively. Alternatively, the pixel can be "filled" until it is "full":

$$C_{FB} = C_{FB} + \min(\alpha_{IN}, 1-\alpha_{FB}) * (C_{IN} - C_{FB})$$

$$\alpha_{FB} = \alpha_{FB} + \min(\alpha_{IN}, 1-\alpha_{FB})$$

The later method can be used to render front facing polygons sorted in front to back order, essentially a painter's algorithm in reverse. This approach is useable with priority based hidden surface removal, without requiring sub pixel bit masks! Remarkably, it renders correctly almost everywhere. [11] The only inaccuracy occurs at the intersection of silhouette edges. Others have noted that the visual effect of this inaccuracy "has not proven to be noticeable." [10]

The relationship between the choice of antialiasing technique and visibility (i.e., hidden surface) technique may be significant. Exact antialiasing requires hidden

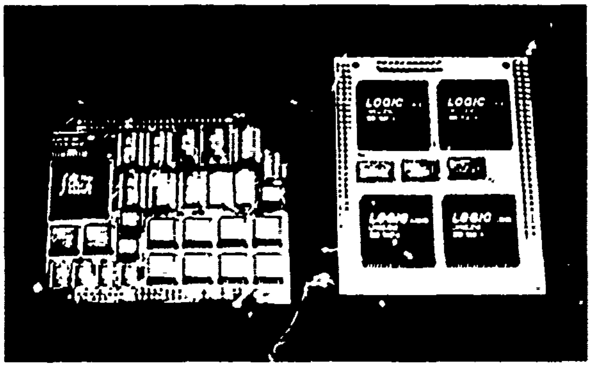


Figure 12. Encoding/filtering module and the Gouraud module.

surface removal before filtering. The most common hidden surface techniques are point sampling methods (cellular priority with sub-pixel bit masks, z-buffers, and ray tracing). The dominance of point sampling hidden surface methods may be a factor in why supersampling is the usual antialiasing technique.

From a cost standpoint, the driving factor is how much circuitry is required to implement shape filtering. Because the intersections required must be calculated whether antialiasing is performed or not, those calculations are essentially free. The real burden of shape filtering is the encoding and the filter tables themselves. Figure 12 shows a photograph comparing two circuit modules used in the system. The module on the left is the encoding/filtering module and the one on the right is the Gouraud shading module. They are roughly equivalent in size and gate counts. Overall, shape filtering requires about 30% more circuitry than no antialiasing at all.

The system as a whole consists of three 9U-sized VME circuit boards. The first board performs the interpolations in the y-direction. The second board performs the interpolations in the x-direction, Gouraud shading, shape encoding, and filter table look-up. The third board contains the frame buffer, alpha blending, and video output. The system runs at 20 MHz and modifies up to 4 pixels in every clock cycle. This results in an overall polygon throughput (when used with a front-end performing hidden surface removal) of 600 thousand antialiased polygons per second (interlaced) or 300 thousand polygons (non-interlaced). The line throughput (averaged over all orientations) is 2 million antialiased 10-pixel vectors per second. This performance is competitive with larger systems requiring extensive custom VLSI.

The simplicity of shape filtering comes from having the most complex part of the calculations, the generation of the filter tables, performed off-line. In other words, for supersampling the computational load is to 1) calculate all the samples and 2) filter them. Calculating many samples for each displayed pixel is the major computational load. With shape filtering the computational load is to 1) calculate the intersections, 2) encode shape, 3) look up the filter result in a table. The intersections have to be computed anyway, so they're free. The circuitry required to encode shape is small and the filter table look-up is a single memory read.

RAMIFICATIONS

While the suggested design is only for lines and shaded polygons, this approach has ramifications for all classes of display and image generator. Shape filtering demonstrates that it is easier to antialias line and polygon displays than is currently practiced. [21] This design allows all display generators, from PC's and workstations to the more sophisticated display generators for Air Traffic Control and future cockpit designs, to have the best antialiasing possible at modest cost.

We also believe that rethinking fundamental algorithms and architectures (as exemplified by shape filtering) is the direction that the design of future image generators must take. Image generation requires many computationally intensive tasks. Antialiasing of polygon edges and lines is one of these tasks (antialiasing of textures and hidden surface removal are two other costly tasks). The universal goals for real-time graphics

- High image quality and performance
- Small size and low cost

cannot be achieved in a reasonable time frame with only the brute-force application of ever more powerful silicon technology to existing architectures and algorithms. These goals can only be attained by the combination of more efficient architectures and algorithms with advancing semiconductor technology.

CONCLUSIONS

Shape filtering eliminates aliasing artifacts with less computational cost than supersampling. Lower cost is usually achieved only with some degree of compromise in quality or performance. That is not the case here. Shape filtering consistently antialiases both lines and polygons with a quality demonstrably at the limit of human visual perception. For supersampling to achieve similar results for polygons alone would require at least 64 samples per input pixel: about an order of magnitude more than what today's image generators can provide.

Shape filtering can be implemented as an enhancement to conventional aliased scan conversion. The incremental cost, in terms of chip counts or gate counts, is about the same as Gouraud shading. In short, though supersampling and shape filtering are both approximations to the exact convolution filter, shape filtering achieves better image quality than supersampling at lower cost. Supersampling is not an efficient approach to antialiasing. It forces a severe trade-off between cost and image quality. The best solution to this trade-off is to avoid it entirely.

REFERENCES

[1] Akeley, K. and T. Jermoluk, "High-Performance Polygon Rendering," SIGGRAPH '88 Proceedings, Computer Graphics, V. 22, No. 4, August 1988, pp. 239-246.

[2] Barkans, A. C., "High Speed High Quality Antialiased Vector Generation," SIGGRAPH '90 Proceedings, Computer Graphics, V. 24, No. 4, August 1990, pp. 319-326.

[3] Blinn, J. F., "Return of the Jaggy," IEEE Computer Graphics and Applications, V. 9. No. 2, March 1989, pp. 82-89.

[4] Bracewell, R. N., The Fourier Transform and its Applications, McGraw-Hill Book Company, New York, 1978.

[5] Brassel, K. E. and R. Fegas, "An Algorithm for Shading of Regions on Vector Display Devices," SIGGRAPH '79 Proceedings, Computer Graphics, V. 13, No. 2, August 1979, pp. 126-133.

[6] Bresenham, J. E., "Algorithm for Computer Control of a Digital Plotter," IBM Systems Journal, V. 4, No. 1, 1965, pp. 25-30.

[7] Carpenter, L. "The A-buffer, an Antialiased Hidden Surface Method," SIGGRAPH '84 Proceedings, Computer Graphics, V. 18, No. 3, July 1984, pp. 103-108.

[8] Cook, R. L., "Stochastic Sampling in Computer Graphics," ACM Trans. on Graphics, V. 5, No. 1, January 1986, pp. 51-72.

[9] Crow, F. C., "A Comparison of Antialiasing Techniques," IEEE Computer Graphics and Applications, V. 1, No. 1, January 1981, pp. 40-48.

[10] Crow, F.C., "Antialiasing" in Computer Graphics Techniques - Theory and Practice, D F Rodgers and R. A. Earnshaw, Eds. Springer-Verlag, New York, 1990.

[11] Gish, W., and A. Tanner, "Antialiased Scan Conversion of Lines and Polygons," submitted for publication.

[12] Gupta, S. and R. F. Sproull, "Filtering Edges for Grey-Scale Displays," SIGGRAPH '81 Proceedings, Computer Graphics, V. 15, No. 3, August 1981, pp. 1-7.

[13] Haeberli, P. and K. Akeley, "The Accumulation Buffer: Hardware Support for High-Quality Rendering," SIGGRAPH '90 Proceedings, Computer Graphics, V. 24, No. 4, August 1990, pp. 309-318.

[14] Kajiya, J. T., "The Rendering Equation," SIGGRAPH '86 Proceedings, Computer Graphics, V. 20, No. 4, August 1986, pp. 143-150.

[15] Krantz, J. H., and L. D. Silverstein, "The Grey-Scale Asymptote for Anti-Aliasing Graphics Images on Color Matrix Displays," Soc. for Information Display 89 Digest, pp. 139-141.

[16] Leler, W. J., "Human Vision, Anti-Aliasing, and the Cheap 4000 Line Display," SIGGRAPH '80 Proceedings, Computer Graphics, V. 14, No. 3, July 1980, pp. 308-313.

[17] Mitchell, D P. and A. N. Netravali, "Reconstruction Filters in Computer Graphics," SIGGRAPH '88 Proceedings, Computer Graphics, V. 22, No. 4, August 1988, pp. 221-227.

[18] Naiman, A. C. and W. Makous, "Information Transmission for Greyscale Edges," Soc. for Information Display 91 Digest, pp. 109-112.

[19] Porter, T. and T. Duff, "Compositing Digital Images," SIGGRAPH '84 Proceedings, Computer Graphics, V. 18, No. 3, July 1984, pp. 253-259.

[20] Szabo, N. S., "Digital Image Anomalies: Static and Dynamic," in Computer Image Generation, B. J. Schacter, Ed., John Wiley & Sons. New York, 1983.

[21] Yan, J. K., "Advances in Computer-Generated Imagery for Flight Simulation," IEEE Computer Graphics and Applications, V. 5, No. 8, August 1985, pp. 37-51.

ACKNOWLEDGEMENTS

The authors wish to thank Dr. Frank Crow of Apple Computer for his permission to use the images in Figure 4. We would also like to acknowledge the continuing support of Lorin Bice, President, and everyone else at Terabit for making this work possible.

ABOUT THE AUTHORS

Walter Gish is Executive Scientist at Terabit Computer Engineering. He is responsible for the development of advanced visual systems. Mr. Gish holds BS and MS degrees in Physics and Applied Mathematics from the California Institute of Technology.

Allen Tanner is the founder and Chief Executive Officer of Terabit Computer Enginerring. He has designed and implemented a wide variety of computer image generation and video conversion systems. Mr. Tanner received his BS in Computer Science from the University of Utah.

AN EVALUATION OF DOME DISPLAY SUITABILITY FOR SIDE-BY-SIDE CREWMEMBER VIEWING

Edward A. Martin
Aeronautical Systems Division
Wright-Patterson Air Force Base OH 45433-6503

ABSTRACT

An evaluation was conducted to determine whether the distortions introduced when a dome display system is used with a cockpit configured for side-by-side seating would significantly impact the operational effectiveness of the simulator. Specifically, the parallax and size distortion which arises due to the dome geometry were addressed. Evaluators generally agreed that there are some problems involved in coping with the distortions produced in the dome display, but that they are not insurmountable. All evaluators thought that the distortion presented to the pilot and copilot with the eyepoint set for the center position (midway between the pilot's and copilot's eyepoints; i.e., the flight engineer's eyepoint when looking out the window) was acceptable. On the other hand, the majority felt that the worst case situation, with the eyepoint set for the left seat (pilot) and viewed from the right seat (copilot), posed significant problems. There were a number of reported symptoms of "simulator sickness"[1,2] incidental to this evaluation effort; these, and some possible reasons for their occurrence, are also presented.

INTRODUCTION

A dome display system is capable of supporting cross-cockpit viewing and a large field-of-view. A major drawback to using a dome is that the displayed image of a distant object can be corrected for only one eye position at a time. Parallax and image size distortion are introduced when the image of a distant object is projected on the surface of the dome, and viewed from any other than the design eyepoint. Domes are clearly applicable, and have been used extensively, for single seat cockpits. They can be -- and have been -- also applied to two-seat tandem cockpits, where the parallax error off the nose is minimized and the crewmember in the backseat tends to be immersed in head-down tasks. However, domes have not been used on simulators with cockpits configured for side-by-side seating due to concerns regarding the effects of the distortion. As a result, there are little available data to support informed judgements regarding the impact of the distortions on crew perception, behavior, or training in side-by-side cockpits -- or the magnitude of the distortion (readily calculated from the geometry) which can be tolerated.

The parallax (or angular) error introduced in a dome display changes with the distance to the object displayed. If an object is actually at a distance equal to one dome radius from the eyepoint, it will be displayed at the proper angle for all observers. As the object distance increases towards infinity, parallax error increases when the object is observed from other than the design eyepoint. Parallax error is illustrated in Fig. 1. A distant object in front of the aircraft should appear directly ahead of both the pilot (left seat) and copilot (right seat). In Fig. 1 the display geometry is shown correct for the pilot's eyepoint. The parallax introduced by projecting the object's image on the dome's surface in front of the pilot is seen to result in the object appearing to the copilot's left rather than directly ahead.

Size distortion is introduced because the pilot and copilot are viewing the same image from different distances. To illustrate, refer to Fig. 1 and consider a distant object to be at a fixed distance directly to the pilot's left; if the cockpit were to pivot nose-left about its center, that object should move to the right across the pilot's field of view while essentially maintaining a constant angular subtense at the pilot's eye. The pilot is situated left of dome center, so the actual size of the image on the dome's surface has to be manipulated to maintain a constant angular subtense at the pilot's eye. Since the copilot's eyepoint is offset from the pilot's by a distance which is not negligible relative to the dome's radius, size corrections for the pilot's eye will not be correct at the copilot's. Therefore, as the cockpit pivots nose-left about its center with the size maintained correct for the pilot, the copilot should perceive the object as growing larger or moving towards the cockpit. Just the reverse would be true for nose-right rotation, i.e., the copilot would perceive the object as growing smaller or receding from the cockpit. Size distortion is treated further under the discussion of apparatus.

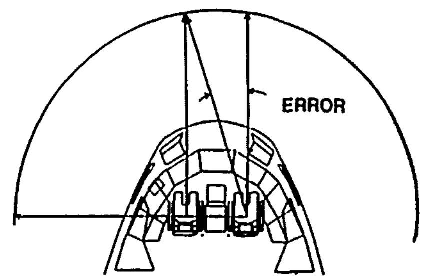


Figure 1. *Parallax error in a dome display system Distant objects forward of the aircraft are shown portrayed correctly for the left seat. Parallax causes these objects to appear to the left when viewed from the right seat.*

A dome display system was selected for the Mission Rehearsal Devices (MRDs) for the Special Operations Forces Aircrew Training System (SOF ATS) because of the attractiveness of the advantages offered by the dome. The cross-cockpit viewing afforded is well suited to support crew coordination activities during mission rehearsals. Each SOF ATS MRD is to be reconfigurable to represent either a C-130, MH-53, or MH-60 cockpit; the dome's large vertical field of view is a cost effective approach for supporting this reconfiguration requirement. The downside was the risk to the program posed by the potential -- and unknown -- impact of the distortions introduced by the dome geometry. Any significant problem regarding the operational use of the MRDs had to be

identified early in the SOF ATS development in order to effectively control this risk.

Timely risk mitigation demanded an empirical evaluation which could be implemented and carried out quickly, using existing facilities and devices where possible. The evaluation was therefore designed to obtain *informed judgements* from the evaluators regarding operational impact; it was not a controlled perceptual or manual control experiment. The purpose of the evaluation was twofold: (1) assess whether any of the distortions produce distracting effects which would *significantly* impact mission rehearsal or training effectiveness, and (2) assess whether optimizing the display for a specific crew position operationally impacts the effectiveness of the display at other crew positions.

GENERAL METHOD

Evaluators

A total of thirty males participated in the evaluation. These fell into three Categories defined as follows: (1) fifteen who are currently flying as aircrew with operational Air Force (C-130 fixed-wing) and Army (MH-60 rotary-wing) Special Operations units, (2) eleven who are experienced C-130 crewmembers, and are now engaged in the management and/or training of SOF aircrews, and (3) four of varying aircrew experience who currently deal with C-130 flight simulator requirements and acquisition. Category (1) is comprised of six pilots with 2200 to 4700 (3600 median) hours in C-130 aircraft, one pilot with 6000 MH-60 hours, and seven Navigators and Flight Engineers with 1350 to 3500 (3000 median) C-130 hours. Of these, only the MH-60 pilot regularly experiences the flight simulator environment. Category (2) consists of four pilots with 2500 to 5700 (4100 median) C-130 hours, and seven Navigators, Flight Engineers, and Electronic Warfare Officers with 1000 to 4300 (2000 median) C-130 hours. The Category (2) members directly involved with aircrew training have about as much current flight experience as the Category (1) members, but regularly experience the simulator environment in addition. Combined, the first and second categories had a median experience level of 3900 hours for pilots and 2500 hours for the other aircrew members.

Apparatus

The evaluation was conducted at the Evans & Sutherland (E&S) facility in Salt Lake City using a prototype of the twenty-four foot diameter dome designed for the SOF ATS MRDs. Three E&S Digital Cathode Ray Tube (CRT) Calligraphic Projectors were configured to provide a three-channel display covering a field of view from 90^0 left to 45^0 right by 40^0 vertical. The projectors in this facility are typically driven to produce a display highlight luminance of about five foot lamberts. Distortion correction (dynamic off-axis correction and nonlinear mapping) was electronically switchable to either the left (pilot's) or center (flight engineer's) eyepoint position. An ESIG-1000 Image Generator and an available visual database for the area around the airfield in Neuburg, Germany, were used to provide the imagery. The terrain around Neuburg is fairly flat and the database had relatively low three-dimensional feature density in this area.

A salvaged C-130 simulator cockpit was installed in the dome, but none of its controls or displays were operational. C-130 aerodynamics were not simulated. Rudimentary flight kinematics were implemented using a 'flybox' through which velocity, attitude, and heading could be controlled. The flybox contained a joystick for attitude control, and a slidelever for airspeed control. Airspeed was displayed in a readout on the flybox. In addition to direct joystick control, heading changes were coupled into aircraft bank. Altitude changes were coupled into pitch. No platform or other type of physical motion cuing was provided.

In the C-130 cockpit, the pilot's eyepoint is 1.75 feet left of center, and the copilot's 1.75 feet right of center. In the geometry evaluated, the design eyepoints are two feet below the dome center. With this geometry, the parallax distortion is as follows. When the display geometry is correct for the pilot's eyepoint, an object image directly in front of the pilot appears 8.5^0 to the flight engineer's left and 16.7^0 to the copilot's left. When the eyepoint is correct for the flight engineer's (center) position, an object image directly in front of the flight engineer appears 8.5^0 to the pilot's right and 8.5^0 to the copilot's left.

Size distortion in the dome configuration used for this evaluation was as follows. With the displayed image size correct for the pilot's position, the visual angle subtended was 26% smaller than it should have been for the copilot viewing a distant object at the extreme left side of the display (i.e., 90^0 left of centerline). At the extreme right side of the display (i.e., 45^0 right of centerline), the visual angle subtended at the copilot's position was 23% greater than it should have been. Similarly, when the displayed image size was correct for the center position, the visual angle subtended at the copilot's position was 13% too small at the extreme left and 11% larger than it should have been at the extreme right. For the pilot, in this latter configuration, the size error varied from 17% too large at the extreme left to 10% too small at the extreme right. From this, it can be seen that if the C-130 cockpit were pivoted nose-right, it would appear to the copilot that the cockpit was moving away from fixed objects in the environment (vice versa for the pilot with eyepoint set at center position) -- or that the C-130 point of rotation was offset some unusual distance from the cockpit center. Evaluator reports suggesting an abnormally offset point-of-rotation (POR), such as "it looked like we were backing up and moving away from the helicopter", were therefore used as indicators of perceived size distortion.

Procedure

The thirty available evaluators were grouped into eight aircrews; operational crew integrity was maintained for those belonging to Category (1). Each aircrew was run through a series of three sessions over a one hour period. Each session consisted of a series of three scenarios which were designed to highlight potential problems in an operationally realistic scene. Each scenario lasted approximately two minutes. The first two scenarios consisted of canned playback flight sequences, with the aircrews acting as passive observers while the aircraft conducted low-level turns over textured terrain and three-dimensional features, an approach to Neuburg Airfield, and a takeoff roll and climb-out from the field. Prior to beginning takeoff roll, the aircraft pivoted nose-right about a point twenty feet behind cockpit center (about 60% of the distance

from the cockpit to the center of the main gear[3]) at one end of the runway; this maneuver was intended to independently display the size distortion of objects (buildings and a helicopter) as they swept across the field of view. During the third scenario, the pilot and copilot 'flew' the aircraft along a pre-briefed route to Neuburg Airfield and landed. This same sequence of three scenarios was repeated for each of three eyepoint positions: 1) correct for the left seat, 2) again correct for the left seat -- but with the left and right seat occupants (pilot and copilot) having exchanged seats, and 3) correct for the flight engineer (center). The eyepoint position sequence was varied from aircrew to aircrew by leaving the eyepoint set to the last position (i.e., left or center) presented to the previous aircrew.

Evaluators annotated and filled out their Evaluation Forms (Fig. 2) as much as possible during the reinitialization periods between scenarios and sessions. All forms were then completed immediately after the one-hour period.

Before commencing, all evaluators had been informed of the purpose and procedures of the evaluation. All evaluators were briefed regarding the events which would occur on each of the two canned scenarios, and the flight path to be followed during the free-flight scenario. Evaluators were encouraged to attempt to fly to specific features or landmarks during free flight, and to have various crewmembers call out bearings to landmarks in order to highlight any crew coordination difficulties introduced by visual parallax. Instructions were also provided regarding the kinds of information desired on the Evaluation Forms (Fig. 2). The intent was to elicit feedback emphasizing effects of distortion, rather than focusing upon the specific nature of the distortion itself; the questions were designed with this in mind. Since the purpose of the evaluation was to determine whether display distortions were of sufficient magnitude to unacceptably impact mission rehearsal or training effectiveness, evaluators were instructed to note the severity of any perceived discrepancies on a scale using the consistent set of descriptors "noticeable, moderate, strong, or unacceptable" to facilitate the identification of trends. The final question on the Evaluation Form requested the evaluator's overall assessment regarding the impact of any observed problems on rehearsal or training value.

RESULTS AND DISCUSSION

The results for each combination of display eyepoint and observer position are discussed in detail below, and summarized in Table 1. For each eyepoint-observer combination, Table 1 provides the total number (and percentage) of observers reporting different severity levels of parallax problems, the number of reports of observable size distortion, and a summary of the overall assessments regarding impact on mission rehearsal or training.

Evaluations with Observer at Correct Eyepoint

This section summarizes the distortion problems reported by evaluators observing the scene from the geometrically correct eyepoint. This provides a baseline for identifying the tendency for observer errors, versus trends which might indicate a serious problem.

Two Category (1), three Category (2), and two Category (3) evaluators observed the display from the center seat, with the eyepoint optically correct for the center seat. Five [one Category (1), two Category (2), and two Category (3)] of the seven evaluators responded that they observed no problems. Two [one Category (1) and one Category (2)] observers noted a moderate parallax effect. No one reported an abnormal point-of-rotation (POR). In their overall assessments, six reported no observed problems, while one Category (1) evaluator was suffering simulator sickness so badly he had to exit the cockpit after the sequence of scenarios for this one eyepoint condition.

Nine Category (1), nine Category (2), and two Category (3) evaluators observed the display from the left seat, with the eyepoint optically correct for the left seat. Twelve [six Category (1), four Category (2), and two Category (3)] of the twenty evaluators noted no parallax problem. Eight [three Category (1) and five Category (2)] evaluators reported noticeable parallax. Four Category (2) evaluators reported an abnormal POR. Seventeen of the overall assessments reported no problems, two [one Category (1) and one Category (2)] evaluators reported minor (but acceptable) problems, and one Category (2) evaluator reported a strong tendency for simulator sickness to be induced.

Some perceptions of visual distortion, when none should exist, could be attributed to the way in which the aircraft flight kinematics were implemented. Reported nose attitude problems, for example, could simply reflect the absence of any angle of attack simulation. The person recording the canned scenarios had difficulty lining the aircraft up on centerline with the flybox; this is a likely source of many complaints of "aircraft right of center" on runway alignment -- a misalignment which appears to have been ascribed to parallax by some evaluators.

It is not clear what contributed to the perceptions of abnormal POR when size distortion should not have been present. None of those reporting this phenomenon were pilots, and so were observing from behind the pilot's seat. However, it is unlikely that this relatively small displacement from the design eyepoint was sufficient to produce a noticeable effect. Assuming that the distortion correction was properly implemented, these reports would reflect observer errors.

Evaluations with Eyepoint Center and Observer in Left/Right Seat

For these two conditions, the eyepoint is optically correct for the center seat, and observed from either the left or right seat. There were ten observers from each seat position. Although right seat evaluators were not the same individuals as left seat evaluators, individuals from the three categories of evaluator populations were about equally distributed across the two conditions. These data are pooled since the reports from either of the two seats tended to be very similar. Ten evaluators were Category (1), nine Category (2), and one Category (3).

Of the twenty total evaluators, ten evaluators reported no problems. There were seven [three Category (1), three Category (2), and one Category (3)] reports of noticeable parallax, two [one Category (1) and one Category (2)] reports of moderate parallax, and one Category (1) report of strong parallax effects (the "strong" effect applied only while landing in the free flight scenario). Only

DOME DEMONSTRATION EVALUATION

	Name__________________Org.____________
Eyepoint	Demo seat (left,center,right)_________
Position #1	Crew Position in Aircraft_____________

Respond to each of the following for eyepoint condition 1,2,&3. Describe the nature and severity of any discrepancies noted. Use the descriptors <u>noticeable</u>, <u>moderate</u>, <u>strong</u>, or <u>unacceptable</u> to describe severity.

1. Canned Scenario #1: Level turns during circling approach, descent, and touchdown.
   a. Do you notice any disconcerting discrepancy or disorientation in your motion relative to the ground during level turns? If so, describe nature and severity.

   b. On final approach and descent to touchdown, does your alignment and motion relative to the runway appear proper? Describe the nature and severity of any disconcerting discrepancy regarding relative direction of the runway or other objects.

2. Canned Scenario #2: Ownship rotating in place on runway, takeoff, level off at low altitude, and level turns over hardened hangars.
   a. Watch the stationary helicopter and surrounding buildings during ownship rotation. Do the helicopter and other objects appear to be moving across your field of view as you would expect? If not, describe the nature and severity of any discrepancy.

   b. During takeoff roll and climb out, does your motion relative to the runway and horizon seem proper? Describe the nature and severity of any disconcerting discrepancy regarding your motion over the runway or relative to objects/buildings alongside.

   c. During level turns over the buildings, do you notice any disconcerting discrepancy or disorientation? If so, describe the nature and severity.

3. Free Flight: Does visual parallax cause disorientation or other difficulty either while you are controlling the aircraft or verbally coordinating object bearings with other crewmembers? Describe the nature and severity of any discrepancy.

4. Overall, do you feel that any of the above noted discrepancies will seriously degrade the mission rehearsal or training value of the devices? If so, discuss their impact on mission rehearsal or training.

Figure 2. *Dome Demonstration Evaluation Form.*

two [one Category (1) and one Category (2)] evaluators reported a noticeably offset POR -- fewer than reported for the left-seat optically correct eyepoint. With the exception of one Category (2) evaluator who suffered moderate simulator sickness symptoms and felt that there was "something unacceptably wrong", there were no overall ratings of "unacceptable". Thirteen [seven Category (1), and six Category (2)] evaluators reported no problems (two actually recommended this configuration), and six [three Category (1), two Category (2), and one Category (3)] evaluators reported noticeable (but acceptable) discrepancies.

Since these same evaluators participated in the correct eyepoint condition, it's interesting to note the small difference in reported problems between the two cases. While these results may suggest that these evaluators did not recognize the distortion present for this configuration, it is also possible that they reflect the sensitivity of the evaluators to the objective of the evaluation, i.e., identify significant problems. There may also be some confounding 'order effects' present; unfortunately, the order in which eyepoint positions were presented to each crew was not consistently recorded with sufficient accuracy to support an analysis for such effects.

Evaluations with Eyepoint Left and Observer in Center Seat

For this condition, the eyepoint is optically correct for the left seat and observed from the center seat. There were ten observers for this condition, with all three categories of evaluator populations being represented [five Category (1), three Category (2), and two Category (3)].

Of the ten evaluators, zero reported no problems. There were three [one Category (1) and two Category (3)] reports of noticeable parallax, six [four Category (1) and two Category (2)] reports of strong parallax, and one Category (2) report of unacceptable parallax effects. Again, only two [one Category (1) and one Category (3)] evaluators reported a noticeably abnormal POR. Four [two Category (1) and two Category (2)] evaluators provided an overall rating of "unacceptable" for mission rehearsal and/or training with this configuration.

The differences in subjective ratings between this condition and the previous is something of a surprise, since the objective magnitude of the distortion should be the same. This could suggest a problem in the distortion correction for the left eyepoint, which would help explain some of the other anomalies associated with this eyepoint.

Evaluations with Eyepoint Left and Observer in Right Seat

In this condition, the eyepoint is optically correct for the left seat and observed from the right seat (the maximum distortion case). There were nineteen observers for this condition. Nine were Category (1), eight Category (2), and two Category (3) evaluators.

Of the nineteen evaluators, zero reported no problems. There were three Category (2) reports of noticeable parallax, three [one Category (1), one Category (2), and one Category (3)] reports of moderate parallax, five [four Category (1) and one Category (3)] reports of strong parallax, and eight [four Category (1) and four Category (2)] reports of unacceptable parallax problems. There were five [one Category (1), three Category (2), and one Category (3)] reports of noticeable, one Category (2) report of moderate, one Category (1) report of strong, and one Category (1) report of unacceptable abnormal POR. In the overall assessments, four Category (1) and three Category (2) evaluators rated this configuration as "unacceptable". Four Category (1), four Category (2) and one Category (3) evaluators felt that this configuration resulted in moderate to serious problems. These included serious degradation of the copilot's ability to backup the pilot, a serious impediment to crew coordination, and the introduction of eyestrain, vertigo and other symptoms of simulator sickness. There was one

TABLE .

Number (and percentage) of observers reporting parallax, size distortion (abnormal POR), and overall mission rehearsal/training impact problems at the various severity levels reported for each display condition

EYEPOINT POSITION: / OBSERVER POSITION:	CENTER / CENTER	LEFT / LEFT	CENTER / LEFT/RIGHT	LEFT / CENTER	LEFT / RIGHT
TOTAL OBSERVERS:	7 (100%)	20 (100%)	20 (100%)	10 (100%)	19 (100%)
PARALLAX SEVERITY					
NO PROBLEM:	5 (71%)	12 (60%)	10 (50%)	0	0
NOTICEABLE:	0	8 (40%)	7 (35%)	3 (30%)	3 (16%)
MODERATE:	2 (29%)	0	2 (10%)	0	3 (16%)
STRONG:	0	0	1 (5%)	6 (60%)	5 (26%)
UNACCEPTABLE:	0	0	0	1 (10%)	8 (42%)
SIZE DISTORTION:	0	4 (20%)	2 (10%)	2 (20%)	8 (42%)
OVERALL ASSESSMENTS					
NO PROBLEM:	6 (86%)	17 (85%)	13 (65%)	1 (10%)	0
NOTICEABLE/MINOR:	0	2 (10%)	6 (30%)	1 (10%)	1 (5%)
MODERATE:	0	0	0	1 (10%)	3 (16%)
STRONG/SERIOUS:	0	1 (5%)	0	1 (10%)	6 (32%)
UNACCEPTABLE:	1 (14%)	0	1 (5%)	4 (40%)	7 (37%)
NO COMMENT:	0	0	0	2 (20%)	2 (10%)

favorable overall assessment even for this 'maximum distortion' condition: one Category (2) pilot felt strongly that parallax was not so bad that the pilot could not compensate and deal with it, and that there would be no serious problem in using this configuration for mission rehearsal.

It is clear that there is a great deal of dissatisfaction with this configuration. The parallax distortion was apparent to everyone and bothersome to most. The magnitude of size distortion (manifested as abnormal POR) also appears to be emerging as a significant problem for this condition.

Simulator Sickness

The Simulator Sickness Field Manual[1] defines the phenomenon in the following way: "Simulator sickness is a form of motion sickness which sometimes occurs in simulators. It may be induced by either physical or visual motion, or by some unusual combination of these two sources of motion information." Symptoms of simulator sickness may range anywhere from eye strain and headache to stomach distress and vomiting. There were a number of reported incidents during the course of this evaluation, although none so severe as vomiting. It was not an objective of this evaluation to assess any tendency of the simulation to induce simulator sickness, and there was no specific request for the evaluators to document symptoms. As a result most reports were verbal and informal, but there was a definite trend. It appeared that the majority of the Category (1) group experienced some symptom, while there was only one reported Category (2) symptom. This is largely consistent with the increased susceptibility observed for aircrew new to simulators,[1] the fact that one Category (1) pilot with a great deal of current simulator experience complained of mild symptoms (i.e., eyestrain and headache) notwithstanding.

Although geometric distortion possibly contributed to eyestrain and other reported symptoms such as vertigo and nausea, it should be noted that there were some reported incidents at the undistorted eyepoint. It is suspected that the simulator sickness symptoms were largely influenced by the specifics of this particular implementation. In order to conduct the evaluation as expeditiously as possible, existing software was used with little modification. One consequence of this was that aircraft kinematics were derived from software designed for use by development engineers, not experienced pilots. Further, a test 'flybox' joystick was used for control inputs, with no force feedback. Aircraft and controller 'feel' were therefore very dissimilar from the aircraft environment in which many of the evaluators were highly (and currently) experienced. In order to run all evaluators through the simulation in the two days available, the length of each session was limited to one hour. The consequent time and equipment constraints resulted in many of the recommended guidelines and general rules for preventing simulator sickness[1] being violated, e.g., the visual display remained active during slew and freeze/reset. While most symptoms were mild, there was one fairly severe incident in which the evaluator had to exit the cockpit after experiencing only one session (at the undistorted eyepoint); it should be noted that this individual was particularly vulnerable, being a member of the Category (1) group and temporarily suffering from other susceptibility factors.[1] This experience underscores the need for close attention and adherence to the published guidelines as visual field-of-view and imagery fidelity are increased -- particularly when the individuals being exposed to the simulator environment are highly experienced, currently operational, and relatively new to the simulator environment. The operational effectiveness of simulators can be significantly compromised if care is not taken to prevent the devices from inducing sickness in the crew, e.g., crewmembers may alter their behavior in order to minimize ill effects.[2]

CONCLUSIONS

The results of this evaluation are encouraging. The dome appears to be an acceptable display device for the side-by-side seating configuration evaluated, if the design eyepoint is properly selected.

The best results were obtained when the 'center-seat eyepoint' was observed from the center seat (geometrically undistorted). As would be expected, there was some reported degradation when this same design eyepoint was viewed from either the left or right seat. On the other hand, there is little difference between the results obtained from evaluators observing the 'left-seat eyepoint' from the left seat (also geometrically undistorted) or observing the 'center-seat eyepoint' from either the left or right seats (geometrically distorted). On this basis, it seems that a 'center-seat eyepoint' is clearly an acceptable design solution to support cross-cockpit viewing.

The worst results were obtained with the 'left-seat eyepoint' observed from the right seat. This was expected since this exhibited the greatest magnitude of distortion. However, the result of viewing this eyepoint from the center seat was expected to be comparable to that obtained when viewing the 'center-seat eyepoint' from either the left or right set due to geometric similarity. This was not the case; the results indicated that the 'left-seat eyepoint' was markedly inferior. While evaluation results do not support use of a 'left-hand eyepoint', it should be kept in mind that the results may be confounded by a problem in distortion correction at the left eyepoint.

A number of occurrences of "simulator sickness" were observed incidental to this evaluation. While some relationship between dome-introduced display distortions and simulator sickness possibly exists, the large number of confounding factors in this evaluation do not permit any conclusion regarding such a relationship to be drawn. A separate study would be necessary to investigate the relationship between dome display distortions and simulator sickness.

As a bottom line, no reason was found to discontinue moving forward with the dome display configuration for the SOF ATS MRDs. At a minimum, it appears that the center eyepoint position could be viewed from any cockpit position without introducing significant problems.

REFERENCES

[1] NAVTRASYSCEN (1989). Simulator Sickness, Field Manual Mod 4. Naval Training Systems Center, Code 2, Orlando FL 32826-3224.

[2] Alderete, T. (1991). Simulator-Induced Sickness: A Status Report. NASA Steering Committee on Simulator Sickness. Flight Systems & Simulation

Research Division, Ames Research Center, Moffett Field CA 94035.

[3] Lambert, M. (Ed.) (1990). Jane's All the World's Aircraft. Jane's Information Group Inc., Alexandria VA 22314-1651.

ACKNOWLEDGMENTS

The author wishes to thank Michael Conquest, now with the Air Force Armstrong Laboratory, for his contributions during the initial development of this paper, and Cheryl Colbert of Loral Defense Systems and Tom Mackowiak of Evans & Sutherland for their constructive critiques and insightful comments in reviewing the manuscript.

ABOUT THE AUTHOR

Dr Edward A Martin is a Technical Specialist in the Training Device Technology Branch, Directorate of Support Systems Engineering, Aeronautical Systems Division, Wright-Patterson Air Force Base OH. He received his BS in Electrical Engineering from the Virginia Military Institute, his MS in Electrical Engineering from Syracuse University, and his PhD in Biomedical Engineering from The Ohio State University. His professional experience includes over twenty-five years' involvement in technology development and transition in research, development, and acquisition of flight simulators, mission rehearsal devices, and training systems.

A NEW CRT PROJECTOR WITH ISOTROPIC EDGE-BLENDING AND DIGITAL CONVERGENCE

Paul Lyon
Evans & Sutherland Computer Corporation
600 Komas Drive
Salt Lake City, Utah 84108

ABSTRACT

Evans & Sutherland has developed a new CRT projector for flight simulators. The approaches used for edge-blending, convergence, and geometry correction are different from those used for previous CRT projectors, providing a low-cost solution to multichannel wide-field-of-view display systems. The projector uses impregnated-cathode nine-inch CRTs to achieve bright and very high-resolution rasters.

INTRODUCTION

Evans & Sutherland has manufactured high-performance CRT projectors for simulation since 1982. Previous projector designs emphasized performance over cost. Recently the performance/cost of image generators has improved and spawned the need for high-performance display systems at a lower cost.

The lowest-priced CRT projectors are commercial units that are mass produced for the consumer market. When this type of projector is used for flight simulation, some system performance is compromised to take advantage of the lower price. Several key features a *simulation* CRT projector would have that a commercial projector would not are:

- Improved resolution
- Greater average faceplate power for increased average brightness
- Built-in edge-blending capability
- Improved video stability over a large dynamic range
- Precision dynamic focus
- Improved convergence and geometry stability
- Improved high-voltage dynamic response and regulation
- Flexible packaging
- Motion-base compatibility
- Field-extend capability
- Custom lens capability

The design objective for the new projector was to provide a more economical yet a high-performance display system with these features. The new projector exploits technology from prior designs and takes advantage of HDTV (high definition television) developments to meet the unique requirements of flight simulation. The primary design goals were to:

- Lower the cost
- Allow the edge-blending of channels, regardless of channel or raster orientations
- Perform edge-blending over several orders of magnitude of image brightness from daylight to NVG conditions
- Use true digital convergence
- Use real-time interactive software algorithms to control the convergence
- Use geometry correction techniques that improve image stability
- Use a single handheld remote control to edge-blend, converge, and correct all projector channels
- Make a modular, light-weight, and rugged system that is easy to package and maintain

This paper elaborates on these goals and discusses the development decisions and compromises made to meet them.

COST REDUCTION

To design a low-cost CRT projector with high-performance, the hardware architecture should take advantage of recent technology trends. HDTV is gaining momentum worldwide and is pushing commercial projection technology toward higher-quality and higher-resolution images. In Japan and Europe, a variety of new nine-inch rectangular tubes with impregnated (dispenser) cathodes have emerged which boast HDTV capability. Some of these CRTs are finding their way into the production lines of commercial displays. A cost reduction goal for the new CRT projector was to avoid expensive custom CRTs and "piggy back" on the emerging HDTV commercial CRT market to take advantage of mass produced tubes.

Calligraphic lights can be useful in flight simulation, but their deflection and focus requirements add considerable cost to a projection system. Since there are no HDTV or other commercial trends relevant to calligraphic lights, and since the current Evans & Sutherland Calligraphic Projector performs this function, it was decided that a "raster-only" projector would be designed. The new CRT projector hardware architecture is specific to raster generation. Other design architectures which accommodate both raster and calligraphic projection by simply changing PC boards in a rack are bulky. These architectures can't take advantage of the compactness a lower-power raster-only projector system can achieve.

The new projector uses resonant raster deflection techniques with HDTV type magnetic components. The coarse geometry correction (for off-axis projection) is incorporated as part of the resonant deflection circuitry. Fine geometry correction and color convergence signals are magnetically added to the deflections signals through a separate yoke (figure 1).

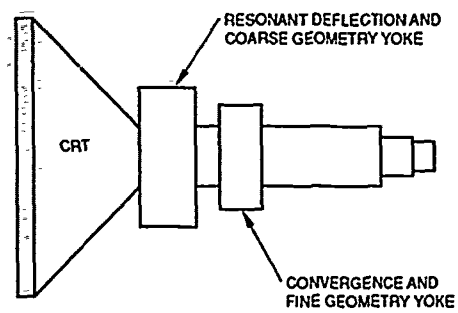


Figure 1. Convergence is magnetically added to the deflection.

By keeping the coarse geometry correction separate from the digital convergence, the required dynamic range of the digital convergence is reduced.

These approaches reduce the number of components and conserve power, thereby lowering the cost of the projector.

ISOTROPIC EDGE-BLENDING

Edge-blending several channels to form a wide field of view is fundamental to the use of CRT projectors in flight simulation. A design goal for the new CRT projector was to edge-match channels regardless of channel or raster orientation. This was accomplished by using polygons to define the blending boundaries and by applying a modified Gouraud shading technique to blend the intensities of adjacent polygons.

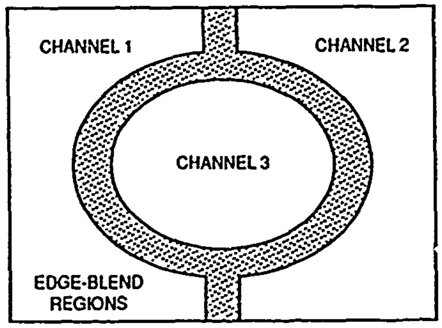


Figure 2. Generalized edge-blending

Figure 2 illustrates one strategy which employs curved channel boundaries and multiple channel overlaps.

Cathode Ray Tubes are nonlinear devices. The light output of a CRT varies as a function of the drive voltage (figure 3). The exponent gamma is usually assumed to be constant; however, at low drive levels gamma varies enough to upset the edge-blending [1].

In flight simulation, channels need to blend together over several orders of magnitude of image brightness from daylight to night vision goggle conditions [2]. The new projector corrects for gamma variations with compensating video circuits and real-time gamma-mapping software.

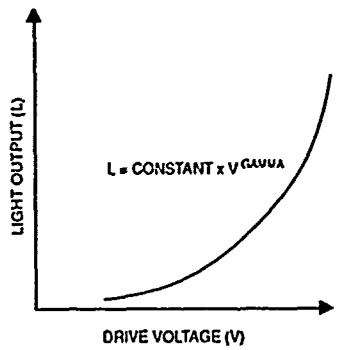


Figure 3. Light output versus drive voltage for a CRT

The video modulation hardware in the projector is simple. A single correction plane is used to map the video shading and edge-match functions for the red, green, and blue CRTs. A videoRAM memory array is used for the correction plane which maps into a three-wide RAMDAC (figure 4).

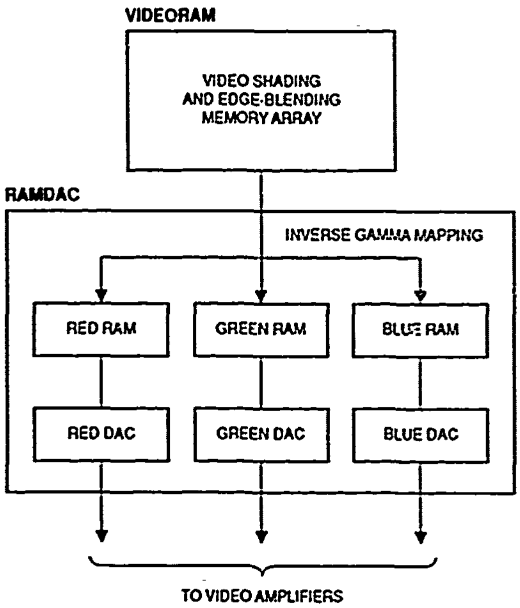


Figure 4. Video modulation hardware

The inverse-gamma mapping for each color is contained in the RAMDAC memories and can be changed in real time. Video shading and edge-blending functions are mapped into screen space instead of pixel space so changes in the sync and timing of a channel won't affect system alignment.

The electrical characteristics of a CRT are sensitive to temperature and can effect the edge-blending as a projector warms up or cools down. The projector's video circuitry incorporates several feedback loops to stabilize the edge-match, especially for low drive levels where variations in black-level have an adverse result.

To achieve high brightness, the CRT faceplates are driven at an average power of 50–80 watts. Even the best phosphors tend to saturate at these current densities (especially blue), so video intensity becomes a function of the electron beam focus. The new projector can precisely control focus to compensate for phosphor saturation and improve the intensity and color match of adjacent channels. To achieve the best focus for various time-of-day conditions, the nominal focus is automatically adjusted as the average scene intensity changes. This is done by the monitoring of the average beam current and the feeding of this information back to the focus control hardware.

DIGITAL CONVERGENCE

Digital convergence is different from digitally controlled convergence [3]. Digitally controlled convergence implies that digital-to-analog converters are used to control analog-derived convergence signals. True digital convergence means that the convergence functions are computed digitally by a microprocessor, stored in a memory, interpolated, and then converted to a signal by a digital-to-analog converter (figure 5).

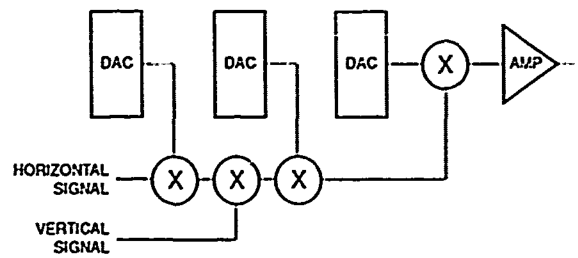


Figure 5a. Digitally controlled convergence

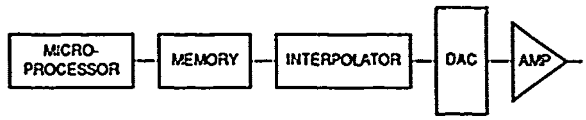


Figure 5b. True digital convergence

True digital convergence is inherently more stable and has the advantage of being flexible. Virtually any kind of convergence function can be realized, and simple user interfaces can be incorporated into the software [4].

Real-time interactive convergence algorithms are important for accurate user-friendly interfaces. The new projector can update the convergence memory in real time. By manipulating and watching the projected test pattern, any modification to the convergence is instantly sent back to the user. Algorithms and hardware that require offline computations or smoothing algorithms, without being interactive in real time (even if the initial input is interactive), are cumbersome and can yield less accurate results [5].

To simplify the convergence process, an array of alignment lightpoints is embedded in the screen surface [5]. These points are tiny optical fibers illuminated by LEDs. These fibers are small enough so that even when used with high-gain screens they are invisible when turned off.

The color of the lightpoints is determined by the LED colors, which can be varied to enhance the user interface. The alignment lights are controlled by the handheld remote control. The remote control fits in the palm of a hand (figure 6).

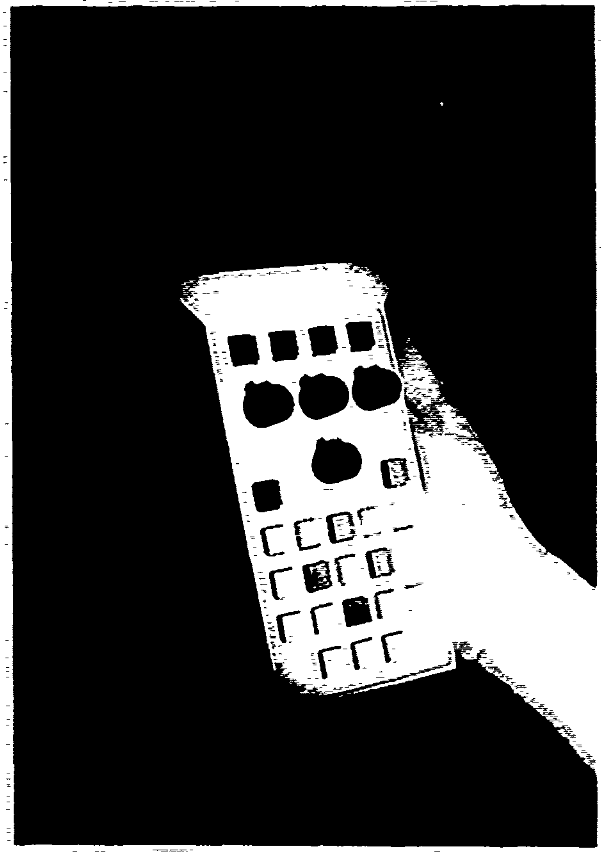


Figure 6. Handheld remote control

The remote control consists of an LCD (with back light), four dials, and several buttons. It uses an infrared or an optional hardwire communications link to the projectors. All projection edge-blending, intensity, convergence, geometry, and focus functions are controlled from the remote control. The user interface is based on popup menus displayed by the projectors [5]. The LCD on the remote control is used for NVG adjustments and other situations where the projectors need to remain dark. All other adjustments are made through the projected menus. This gives user-friendly heads-up control over the display. The system design goal of a single remote control is satisfied by interconnecting the projectors so one remote control can communicate with all projectors.

GEOMETRY CORRECTION

The geometry of a display system refers to the positioning of the projectors relative to the projection surface and the viewer (figure 7).

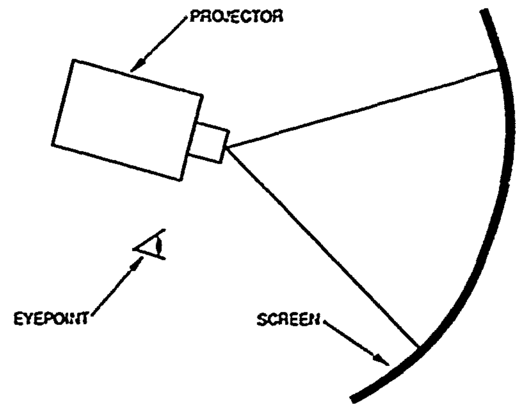


Figure 7. System geometry

The raster of the projectors can be shaped to compensate for the inherent keystone, linearity, and other distortions caused by off-axis projection onto a curved surface. This shaping, or *geometric correction*, is large compared to the correction required for convergence. The geometry correction for the new CRT projector has been incorporated in the resonant deflection circuitry to save power and reduce cost. This coarse correction compensates for most of the distortion, while the fine correction is done with the digital convergence hardware.

A design goal was to improve the stability of the geometry correction. Because of the analog nature of the coarse geometry correction, extra care was taken to preserve stability. The deflection circuits employ closed-loop or feedback designs. Components that are very temperature sensitive, such as linearity coils, are replaced with closed-loop amplifiers. Other temperature compensation circuitry is included to further enhance stability.

The projector has auto-scan or auto-sync capability. It senses changes in the sync signals and automatically adjusts the convergence and geometry to match the new system timing. Both standard and nonstandard sync and timing formats are supported.

MODULARITY AND FLEXIBILITY

A system design goal was to make a modular light-weight projector system that is rugged and easy to package. Figure 8 shows a projector arrangement where both inline and delta CRT configurations are clustered together to optimize the overall system performance. When positioning and packaging projectors to optimize system parameters like viewing volume or moment of inertia, a modular and flexible design is essential.

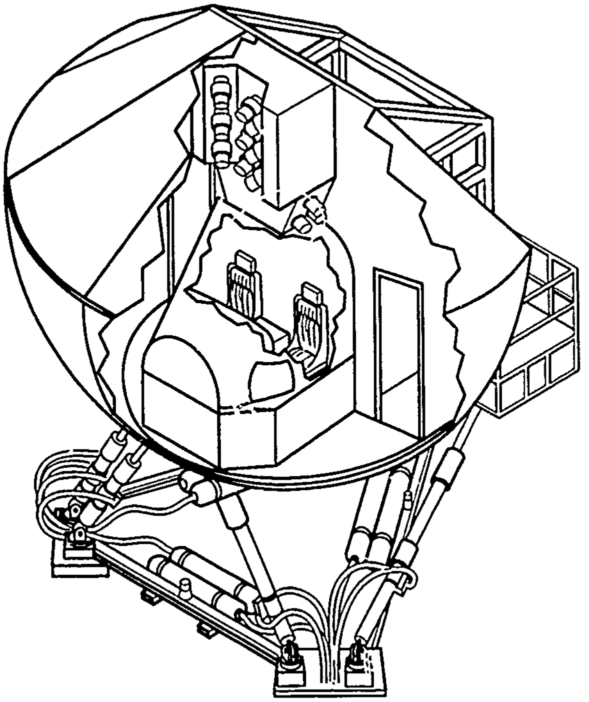


Figure 8. Typical projector configuration

Standard projector packages can't always meet the position and weight-distribution requirements imposed by lenticular screen materials and motion bases. To manage these kinds of system constraints, the new CRT projector was modularized (figure 9). There are three CRT assemblies (red, green, blue), consisting of the CRTs, magnetic components, video amplifiers, and liquid cooling cells. There is a compact electronic control module (ECM), which contains the deflection and correction hardware, digital convergence hardware, and low-voltage power supplies. The high-voltage supply is a separate module.

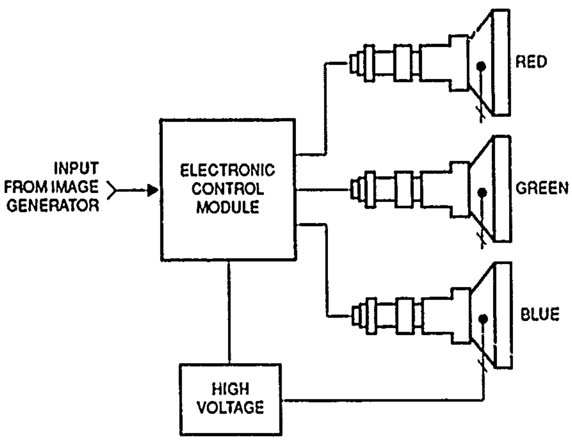


Figure 9. Hardware modules for each channel

Point-to-point cables have been used between modules to simplify the interconnection and to lower the cost. Each module is small and light weight. These three basic modules, combined with the simple interconnection scheme, allow adequate flexibility to accomplish complex packaging requirements.

Parameter	Specification
Power source	110/120 V ac, 208/220 V ac, 50/60 Hz
High voltage	29-35 kV (improved dynamic response and regulation)
Projection tubes	Nine-inch high-brightness CRTs with impregnated cathodes
Faceplate power	50–80 watts continuous
Faceplate cooling	Integral liquid cooling cells
Lenses	Double focus f1.0 (others and custom available)
Luminous output	600 Lumens peak white (f1.0 lens)
Resolution	Over 1,000 TV lines
Convergence accuracy	Less than 0.1% of picture height
Convergence drift	Less than 0.1% of picture height per 24-hour period
Distortion	Less than 1.0% of picture height
Focus	Electromagnetic with average beam current compensation
Horizontal auto-scan	15–36 KHz interlaced, optional 30–72 KHz noninterlaced
Vertical auto-scan	25–120 Hz
Sync	Composite, sync on green or separate H and V sync
Edge-blending	Compatible with daylight through NVG conditions
Convergence	True digital
Geometry correction	Keystone, linearity, and spherical correction
Control	Infrared handheld remote control
Motion compatibility	Operation at 1 G (0–10 Hz), maximum 5.0 G

Table 1. Performance specifications

PERFORMANCE

Table 1 summarizes the performance specifications of the projector. In addition to these specifications, the new projector is capable of a "field extend" function, which makes it compatible with image generators that use this feature for scene overload management. It also incorporates various fault-monitoring and protection circuits to guard against phosphor burns and damage to the electronics. The projector also has a noninterlace option to support image generators capable of the faster horizontal and pixel rates.

SUMMARY

Evans & Sutherland's goal was to develop an economical high-performance CRT projector. The projector is designed for flight simulator applications and includes features not available in commercial projectors. By designing a raster-only projector using HDTV technology, high performance was achieved at a lower cost. The new projector offers generalized edge-blending capable of matching channels over the full range of daylight and nighttime scenes as well as NVG conditions. It uses true digital convergence and incorporates geometric correction as part of the resonant deflection hardware. The projector is modular and light weight to allow flexibility in positioning and packaging.

REFERENCES

1. Lyon, Paul. Edge-Blending: Multiple Projection Displays on a Dome Surface to Form Continuous Wide Angle Fields-of-View. *Proceedings from the 7th IITEC Conference*, November 1985.

2. Green, Mark, and Grayston, Lane. Displays for NVG Flight Training. *Proceedings from the 11th IITSC Conference*, November 1989.

3. Lyon, Paul and Black, Steve. 1984. A Self-Aligning CRT Projection System with Digital Correction. *SID Digest* 15:108–111.

4. Lyon, Paul. A Wide Field-Of-View CRT Projection System with Optical Feedback for Self-Alignment. *Proceedings from the 5th IITEC Conference*, November 1983.

5. Green, Mark and Lyon, Paul. 1988. A New Computer-Human Interface for Aligning and Edge-Matching Multichannel Projector Systems. *SID Digest* 19:109–112.

ACKNOWLEDGMENTS

The author would like to thank the engineering staff and management of Evans & Sutherland Computer Corporation for their hard work and support throughout this project.

ABOUT THE AUTHOR

Mr. Paul C. Lyon is a principal display systems engineer at Evans & Sutherland. He has been involved in various display systems designs since 1982, including both CRT– and light-valve–based systems. He has been involved in the design and development of advanced CRT projectors since 1980. Mr. Lyon holds Bachelor's and Master's Degrees in Electrical Engineering from the University of Utah.

WHY SIMULATORS DON'T FLY LIKE THE AIRPLANE - DATA

William G. Schweikhard, Senior Vice President
and
Daryl J. Schueler, Aerospace Engineer
Kohlman Systems Research
319 Perry Street
Lawrence, Kansas

ABSTRACT

The hardware and software technology for simulators and flight training devices have advanced enormously over the past ten years. We have been able to create very realistic visual scenes with high resolution, brightness and field of view; motion systems that provide the cues that give the feeling of actually flying in the airplane; high fidelity sounds that represent the operating environment inside the airplane; and computers that are capable of mathematically modeling the equations that represent the various components and systems being simulated. The quality of the data that is used to mechanize the flight dynamics and systems of the airplane being simulated is lagging. This paper focuses on the traditional approach for generating simulator design and verification data, and then describes a flight test approach for improving the quality of the data. Data developed by the traditional approach are compared with data developed by the flight test approach. Comparisons are made of simulated versus flight test results for operational maneuvers, one employing traditional data and the other employing flight test generated data. The need for high quality flight test data that exceeds those of current Development Test and Evaluation (DT&E) and Operational Test and Evaluation (OT&E) results is emphasized.

INTRODUCTION

The hardware and software technology for simulators and flight training devices have advanced enormously over the past ten years but we tend to take them for granted. It is well to remind ourselves of the things that have been accomplished, the basics of what we are about, and the opportunities for improvement that still remain.

As for the basics we might ask ourselves ... What is a Simulator?... And one might answer somewhat facetiously:

1. A hoax foisted against the pilots/operators to make them believe they are flying/operating the "real thing"; or more seriously we might answer

2. A unique combination of hardware, software and data which when combined in the proper way creates a realistic illusion of flying/operating the real thing.

The key words here are:

"The real thing" (and I don't mean Coke)
"Illusions"
"Hardware, software and data"

The goal of creating the illusion of the "real thing" leads us to the consideration of the quality of the hardware, software and data. These components are essential, especially the data, to create this illusion. It is the integrated effect of the hardware, software and data which provides the cues for a flight simulator's Cockpit, Instruments, Controls, Sound, Vibrations, Smells, etc. Each of these must progress together. If one is deficient, the quality and acceptability of the simulator suffers.

In the broadest sense we might say that the simulator is acceptable if the analogy holds which says: If it looks like a duck, waddles like a duck, quacks like a duck, swims like a duck, flies like a duck, then it must be a duck. The unfortunate part of this analogy is that it is qualitative and acceptability for today's simulators requires that quantitative criteria be met as well. In fact, if all of the criteria for simulator acceptability could be quantified, it would be possible to build a nearly perfect simulator that is limited only by the physical constraints of the cueing of the hardware. First we must examine the criteria for acceptability.

In the past a simulator was considered acceptable if it met the expectations of the trainer and trainees, whatever that may have been. Generally, it meant that the simulator operated pretty much like the airplane, at least to the point that there were no major distractions that detracted from learning IFR procedures. Over the years the hardware and software have improved to the point of creating some pretty realistic illusions. The expectations of the users have increased as well. "Close" is no longer good enough. If the majority of the training is to be done in the simulator, it must be as exact a replica as possible. Data plays a strong role in achieving this goal. Without good data the best of hardware is just a poor imitation.

SOURCES OF SIMULATOR DATA

Data for yesterday's simulators have come from a number of sources, but primarily it has come from the manufacturer's design data (aerodynamics, propulsion, controls, avionics systems, analytical predictions, weight, center of gravity and inertias, etc.). The problem was integrating the myriad of detailed factors from a multitude of sources into something that makes

sense in terms of the flying qualities and the illusions that were desired. The result was not always the best. A classic example of how the detailed derivatives that make up the stability and control of an airplane can be distorted is illustrated in Figure 1 from Reference 1. Here an erroneous value of C_{l_β} was taken as correct at high angle of attack, and the other parameter C_{l_p} was distorted to compensate in such a way as to replicate the maneuver being matched. But the effect on other flying qualities not being matched is devastating and can result either in negative training or simulator compensation on the part of the pilot if he/she recognizes the model as being wrong.

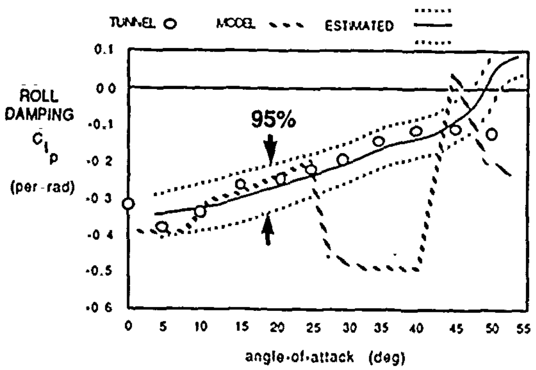


FIGURE 1a. Comparison of roll damping from a variety of sources

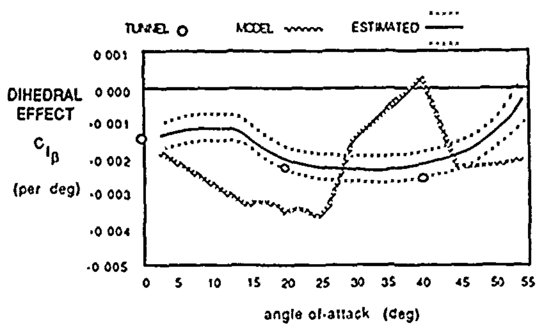


FIGURE 1b. Comparison of Dihedral Effect From a Variety of Sources

An example of the discrepancies that are common when using the data from multiple sources straight away is illustrated in Figure 2a. Here, the values of the various stability derivatives were taken from the manufacturers aerodynamic report for a business jet airplane and compared with the flight test data. The simulator model is being driven by the control surface displacements of the flight tests. When parameter identification techniques are used to determine the stability derivatives for the maneuver, the match is seen in Figure 2b to be identical to the flight test results. These results suggest that simulator data should be derived from flight test but as will be shown this approach is not the whole answer either.

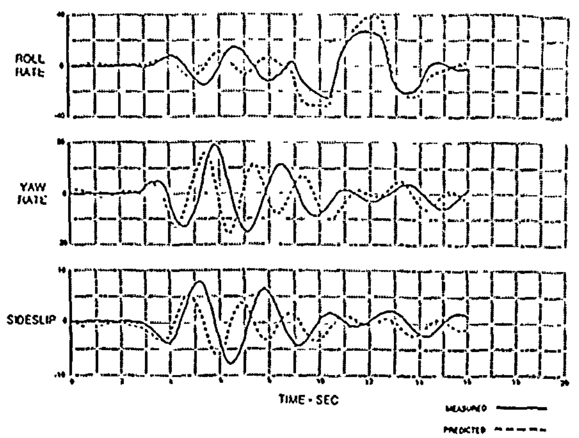


FIGURE 2a. Rudder-Aileron Doublet Predicted Model Response (Mach = .38, Alt. = 10,000 ft.)

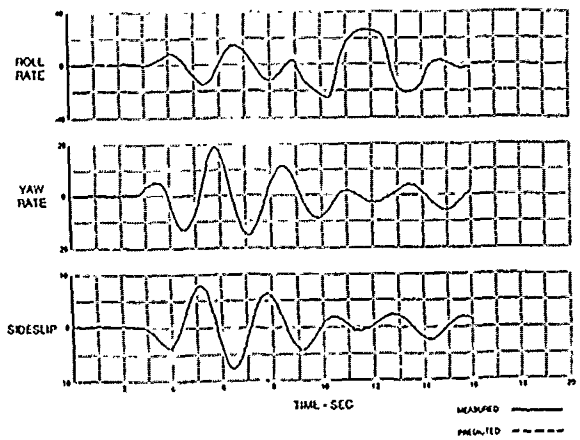


FIGURE 2b. Rudder-Aileron Doublet Final Results After 22 Iterations

FLIGHT TEST GENERATED SIMULATOR DATA

There is a sound rationale for developing simulator models from flight data. The manufacturer can tell you how the airplane is supposed to work. The airplane can tell you how it really does work providing you have the right tools and use them properly.

The benefits of creating a simulator model from flight test results are many compared to the traditional method of using manufacturer's data. First, an aircraft can be chosen that is representative of those operating in the fleet. This fact is especially important as aircraft age and the mechanical components show some slop and wear. Examples of having an airplane representative of the fleet will be given later.

Second, validation data can be collected from the same source as the design data for the simulator. Thus, there is a consistent set of data which, if of high quality and if analyzed properly, can produce very creditable results and a high quality simulator.

Third, the engineering process of parameter identification nulls out the effects of errors, ever present in the physical mass characteristics of the airplane (weight, center of gravity and inertias), because the same values are used in the flight testing as are used in the simulator. It is a happy case of "right or wrong, be consistent."

It must be remembered though that the flight coefficients and derivatives may differ from the wind tunnel values, and that the flight test aerodynamic data and physical data are a matched set and cannot be interchanged with data from other sources (e.g. wind tunnel and computational predictions).

Fourth, aeroelastic effects are an integral part of flight derived coefficients and derivatives. Unlike the wind tunnel model, the airplane is not rigid so it flexes depending on the flight condition, mass distribution and the dynamics of the maneuver that is being analyzed. Thus, aeroelastic effects are buried in the coefficients and derivatives derived from flight tests. Aeroelastic corrections should not be applied when flight test data is incorporated in the simulator model.

Sometimes in using the flight test method we learn things that even the manufacturer didn't discover. This fact is borne out by comparisons of manufacturers design data with flight data.
(Reference 1)

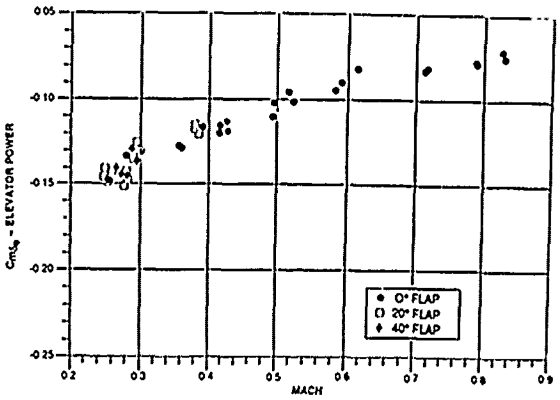


FIGURE 3. Results of Dynamic Maneuvers to Determine Elevator Power by using MMLE Methods

Figure 3 illustrates a set of data points for the elevator power derivative, $C_{m_{\delta_e}}$. Each point is the result of one time history match of a maneuver, like that of figure 2, at a given Mach number. The scatter is typical of that obtained with most flight derived derivatives. Note that this figure includes both flaps down and flaps retracted data. Figure 4 shows a comparison of these flight derived MMLE data, after fairing, with the manufacturers predicted values. The fifty percent error seen here is certainly not good enough.

Another example is shown in Figure 5, which is a comparison of aileron power, $C_{l_{\delta_a}}$, from flight test and the manufacturer's aerodynamic summary. It is clear that significant differences in magnitude and shape may exist between MMLE flight test results and predicted values.

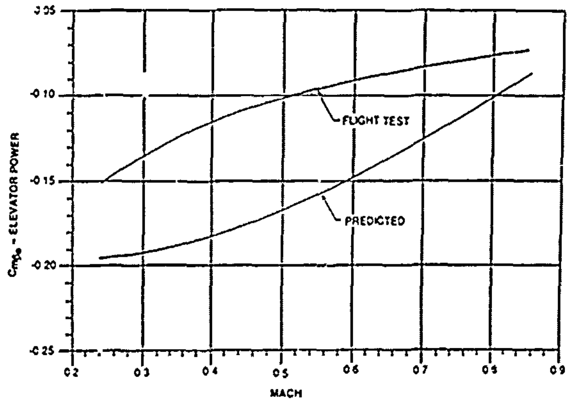


FIGURE 4. Comparison of Flight Test (MMLE) and Predicted Values of $C_{m_{\delta_e}}$ for a Business Jet

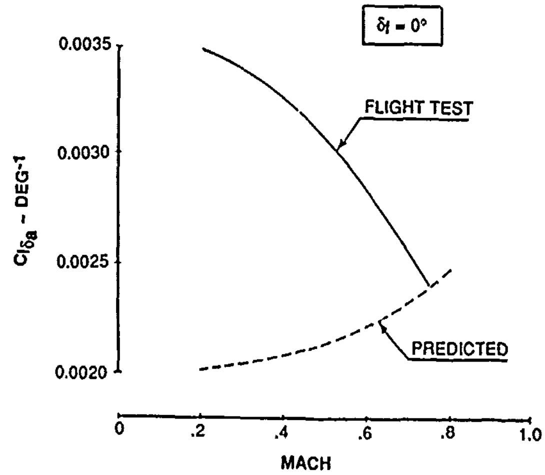


FIGURE 5. Comparison of Flight Test and Predicted Values of Aileron Power, $C_{l_{\delta_a}}$

An area in which flight test parameter ID methods have been particularly useful is in the determination of the effect of quasi-steady aeroelasticity on stability and control derivatives. Figure 6 shows the result of extracting elevator power, $C_{m_{\delta_e}}$, of a popular business jet over a large range of Mach number and dynamic pressure. Elevator deflection was measured near the actuator rod at the root of the elevator. Elevator twisting at high dynamic pressure causes a significant attenuation in elevator power, resulting in important departures from the data produced using relatively rigid wind tunnel models.

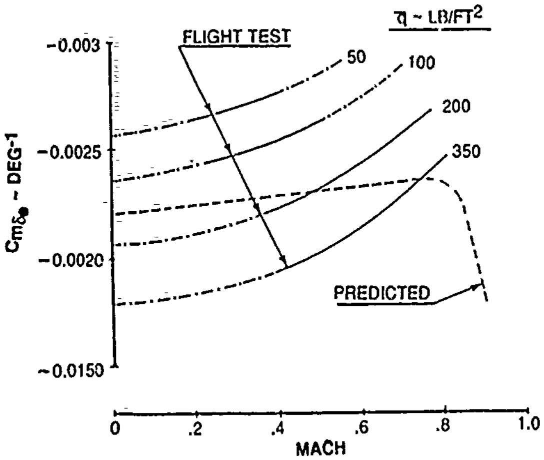


FIGURE 6. Comparison of Flight Test and Predicted Elevator Power with Aeroelastic Effects Included

Another similar example is shown in Figure 7, which shows a data comparison of the dihedral effect, C_{l_β}, of an airplane with winglets. Although the magnitude and shape were well predicted for a rigid airplane, high dynamic pressure causes aeroelastic bending of the winglets, which attenuates the dihedral effect. Such changes must be properly accounted for in simulators.

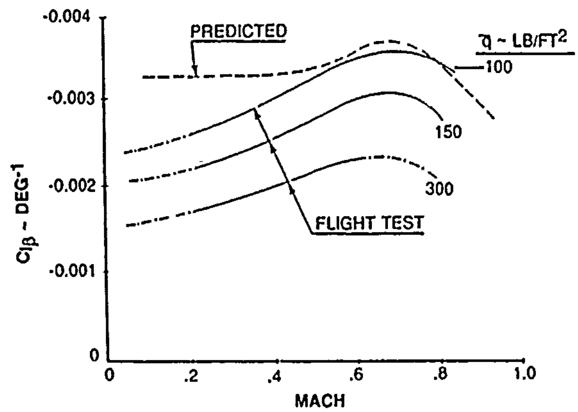


FIGURE 7. Effect of Aeroelasticity on Dihedral Effect, C_{l_β}, of an Airplane with Winglets

Some derivatives like yaw damping due to roll rate, C_{n_p} and rolling moment due to yaw rate, C_{l_r} have been difficult, if not impossible, to extract from flight tests before parameter identification was developed. Now it is relatively straightforward to determine these cross-derivatives, with reasonably good accuracy as shown in Figures 8a, b.

Finally, Figure 9 shows the resolution capability of parameter identification methods. Thus, even when predictive methods are relatively good, effects of secondary parameters, such as angle of attack and flap deflection, can be accurately determined from flight test data.

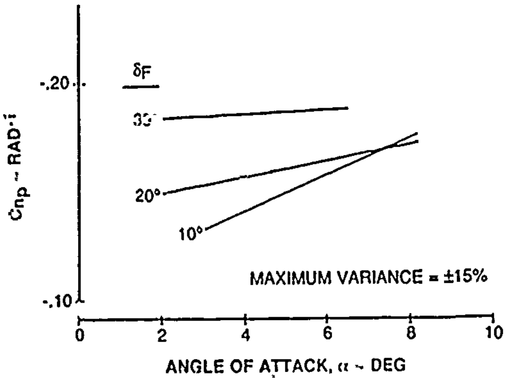


FIGURE 8a. Lateral-Directional Cross Derivatives C_{l_r} and C_{n_p} Determined from MMLE Maneuvers

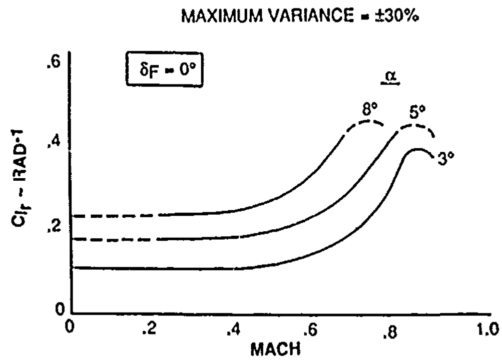


FIGURE 8b. Rolling Moment Due to Yaw Rate Determined from MMLE Maneuvers

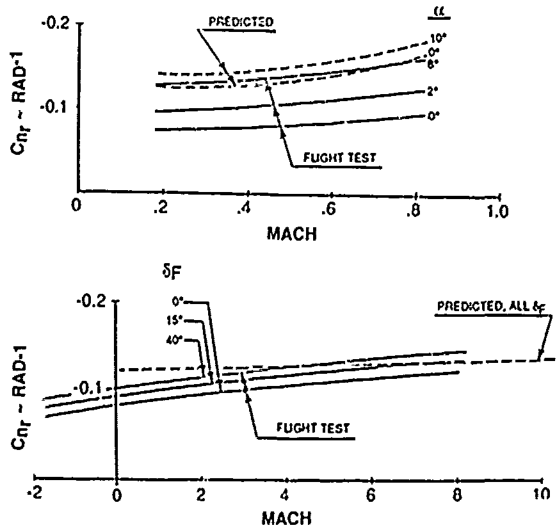


FIGURE 9. Comparison of Flight Test and Predicted Values of Yaw Damping Derivative, C_{n_r}

UNSIMULATED EFFECTS

There is strong evidence of unsimulated effects and derivatives in flight test data. Simulator models include all of the primary derivatives and some of the more commonly encountered secondary derivatives but others are frequently missed especially if the design data model has been derived from maneuver sets that were not designed to reveal them (e.g. a systematically intermeshed combination of Mach number, dynamic pressure and angle of attack).

The implication of the data scatter of Figure 3 is more than simple scatter that is normally found in experimental data. For, if the derivative set for any one test maneuver of Figure 3 is placed in the flight simulator, the match is nearly identical as shown in Figure 10. This figure shows three sets of time history results: the flight test, the match of the flight test using the MMLE model that generated the derivatives, and the flight simulator results using the MMLE derivative set obtained from the flight test data. Both the MMLE and flight simulator models were driven by the flight test control surface deflections for the maneuver. The minor discrepancies between the flight and the model results could be due to small experimental measurement errors in the calibration for the flight control surface deflections of the airplane or to unrecognized and therefore unsimulated terms in the models that produce subtle differences in the results. Note that the match is less exact late in the maneuver during the free response and recovery portion when

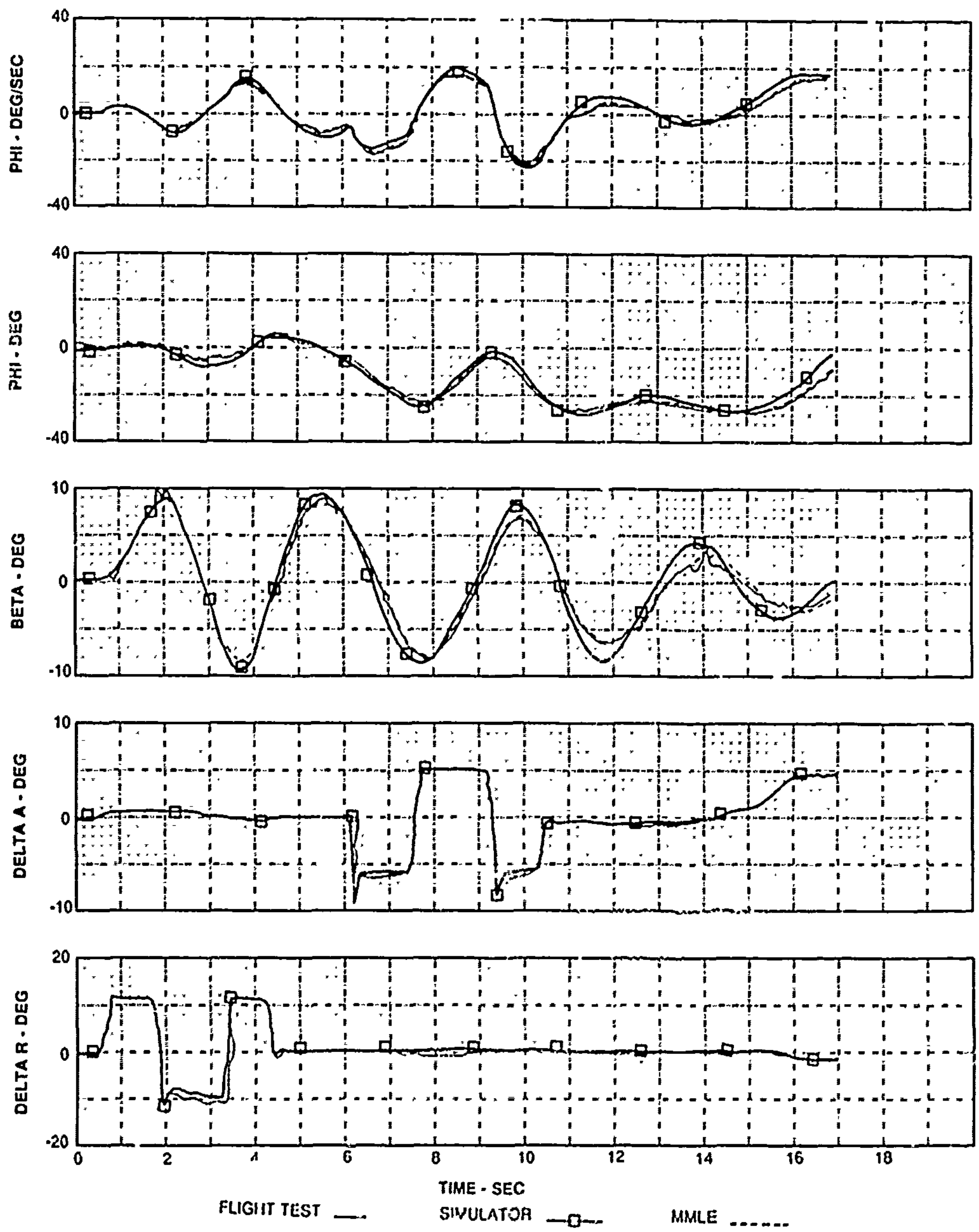


FIGURE 10. Comparison of MMLE Model, Simulator Model and Flight Test using MMLE Derivative Without Tuning the Simulator Model

the magnitude of the disturbances is small. Frequently these discrepancies are attributed to minor errors in the derivative set and the drifts they cause as the integration duration becomes large. However, they could also be caused by the relative magnitudes of the simulated variables compared to the unknown and therefore unsimulated parameters that should have been included in the models.

Thus, the scatter of the data points of Figure 3 may not be scatter at all, but rather, minor differences in modeling where unsimulated and perhaps non-linear parameters are concerned. It should be noted that the data points of Figure 3 that are at the same Mach numbers were done within one minute of each other so the center of gravity and moments of inertia were essentially identical. Yet the derivatives that were derived were slightly different.

These differences as indicated by the scatter become very large for secondary derivatives and for primary derivatives when that mode of motion has not been excited. For example, one would not expect to extract good lateral directional derivatives from a longitudinal maneuver where the lateral-directional motions simply have not been excited.

An example of the effects of variables that are not normally present, and thus are not modeled for most simulators, are shown in Figure 11 for a C-141 airplane. Its configuration, its age and perhaps its aeroelasticity create some anomalous behaviors. Here, a simple rudder doublet has excited both the lateral-directional and the longitudinal modes. The pitching oscillation is twice the frequency of the lateral-directional. A derivative for pitching moment due to sideslip (C_{m_β}) had to be added to the model to properly reproduce the observed motion. Also present in this maneuver is a non-linear floating aileron trim tab oscillation condition and a non-linear asymmetric out of phase thrust variation that contributes to the resulting motion of the airplane. The trim tab slop of almost one half degree is probably a minor effect but the sawtooth and out of phase thrust variation due to sideslip indicated by the EPR1 (left outboard engine) and the EPR3 (right inboard engine) variations are very large, 800 pounds for engine one and 600 pounds for engine three. The total moment produced for all engines is of the order of 25,000 foot pounds which is not of secondary magnitude.

The interplay of these and perhaps other secondary effects, cause the lateral directional oscillatory modes to be unstable at higher altitudes and quite stable at low altitude. A detailed analysis and discussion of these characteristics are beyond the scope of this paper and will have to wait for another paper on another day. But the fact remains that there are other derivatives and effects that are not represented in today's simulator models.

It is the author's belief that the primary and secondary contributors to the motions of an airplane can best be identified through the flight test approach along with some very perceptive analysis of the data by some very skilled interpreters. Some very high quality instrumentation and advanced flight test procedures are required to reveal what might otherwise be missed by other traditional flight test technicians.

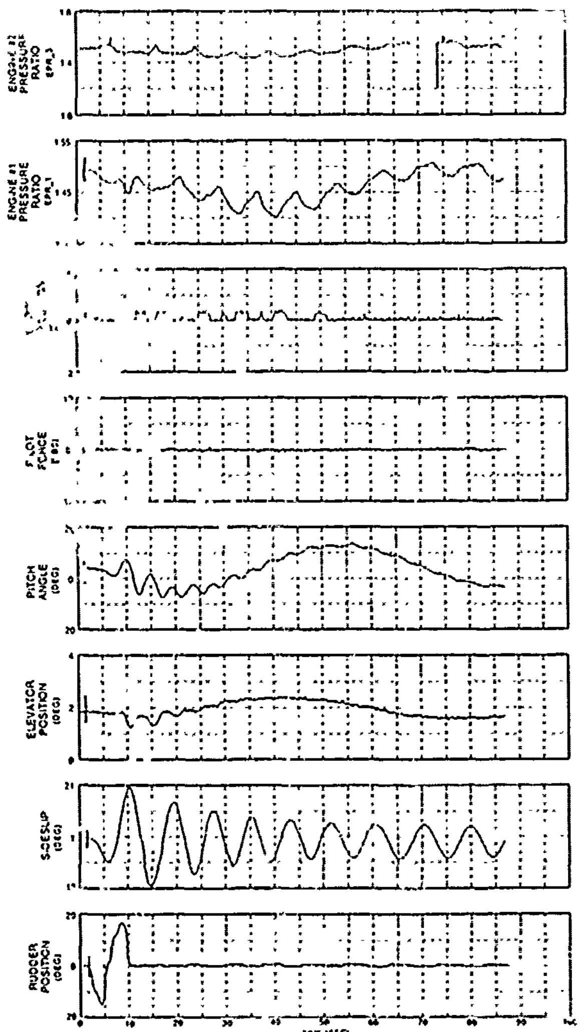


FIGURE 11. Lateral/Directional Controls Free Oscillation of a C-141 Airplane, 29,000 ft.

FLIGHT TEST DATA

Traditional flight test data that is done for development, test and evaluation (DT&E) in the military sector and for FAA certification in the civil sector is none too good for the validation of flight training simulators. Large development and certification programs frequently are done using several test articles each having different purposes (e.g. performance, stability and control, propulsion, avionics, loads and structural dynamics). Each has different types and numbers of sensors to accommodate the specific test areas and objectives assigned to it. Rigorous consistency of type and quality of measurements between the various test articles is not a high priority.

In some cases the various test articles from which the simulator validation data comes are not of the same lineage. For example data may come from a prototype, a preproduction airplane and a production

airplane which are similar but different in ways that do not affect the development process but do seriously impact the simulator acceptability. Data from the power plant and flight control systems are often significantly different between prototype and production airplanes. Yet the data set provided to the simulator manufacturer frequently consists of an unholy mix of data from a number of different aircraft having similar but different characteristics of the various components and systems. Even when all of the different test articles are of the same lineage (i.e. all production airplanes) there are differences in data acquisition systems and the quality of a particularly key measurement from one airplane to the other. The differences are far more important for simulator validation than for developmental testing. Since developmental testing requirements drive the quality of the data gathered during the early stages of testing and even later, much of the simulator data gathered today is inadequate either because of the way the test was flown or the quality of the data taken.

A classic example of the pitfalls of using "best available" DT&E data taken from similar but different test articles is described on reference 3 for the AH-1W helicopter. Data were available from a prototype (DT-11F) and a production (DT-111) aircraft. The prototype data were to be used as the primary set and holes (tests not available) were to be filled with data from the production aircraft. Almost every possible ill was present in these data: noise, data scatter, data shifts, uncertain parameter scaling, missing parameters, uncertain fight conditions, data trend differences, misunderstood pilot techniques, wide test tolerances, etc. Many of these ills are seen in Figure 12a which shows large scatter and poor trend information coming from the two test articles DT-11F and DT-111. The simulator, though not yet finely tuned showed a trend similar to the DT-11F results. A consensus was reached with the test and simulator procuring organizations that the curve fit along with the wide tolerance shown in figure 12b should be the standard for simulator acceptance. For a full discussion of the sort of problems that are encountered when forced to use data that was not specifically intended for validating simulators, the reader should review reference 3 for further details and horror stories. The term "Best Available Data" is frequently used in the simulator validation business. The acronym for this term aptly describes this type of data, BAD. Using best available data is like finding something in the garbage dump, it usually stinks.

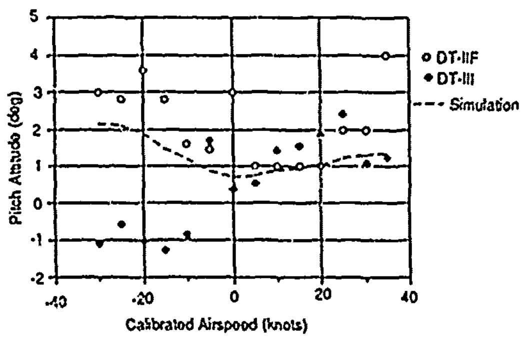


FIGURE 12a. Pitch Attitude Trend Disparity

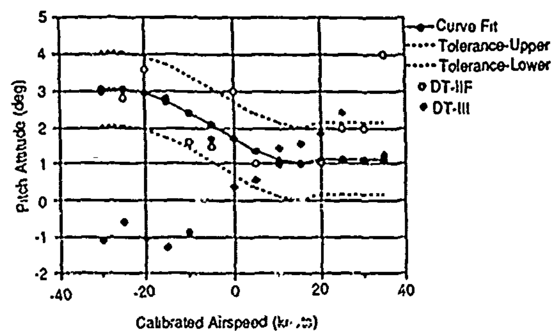


FIGURE 12b. Pitch Attitude Final Test Criteria Definition

Simulator data requirements have redefined the term "Quality Data" with standards that exceed the best available flight research data. Quality data has no noise (< .1%), high precision (accuracy < .05%, resolution < .025%) and has inherent stability (repeatability of .05% and no zero shifts). Great care is required to see that slight zero shifts, calibration changes, flight test techniques and analysis methods are "the best possible", because "the best possible" are sometimes still not good enough. Special flight tests should be run to obtain design and validation data for high quality flight simulators that fly like the airplane.

SIMULATOR VALIDATION PROCESS

As stated at the outset of this paper, the acceptability of simulators a decade or two ago was by and large qualitative, especially in the flying qualities area. The objectives were training in instrument flying procedures and there were no credits for visual takeoff and landings. These procedures were done in the airplane.

The demand for full flight simulation and training has changed this situation. Acceptability has taken on the meaning of truly representing the airplane. The validation process has changed from one of "being close" to one of being as exact a replica of the real thing as possible. Quantitative criteria have replaced much of what was once a qualitative process. These quantitative criteria can be specified in terms of the frequency, damping and time constants of dynamic maneuvers or the more sophisticated frequency domain parameters can be specified. The leadership in this area has come from the civil sector (FAA, IATA, etc.) which embraces the concept of using flight test measured control inputs to drive the simulator controls, then measuring the outputs of the simulator and comparing them with the flight test responses. The differences between the two are evaluated against established tolerances within which they must fall. This provides an end-to-end check of a variety of different types of maneuvers over the flight envelope of the airplane. This process then serves as the basis for requalifying the simulator at regular intervals (once a year for FAA AC 120-40B). This process is described in reference 4.

The validity of this approach assumes that good flight test data and well flown maneuvers are available.

Matching of time history maneuvers is not a random "tweaking" process. Trading of the values of one derivative in order to compensate and match the time histories is what caused the disparities of stability derivatives illustrated in figure 1. Changing C_{l_p} to accommodate an erroneous C_{l_β} created a reasonable match, but two wrongs don't make a right. The simulation had to be lacking in another area where these derivatives are important (e.g. roll characteristics and steady sideslips). Erroneous adjustments then cascade into other parameters that must then be adjusted. Those who successfully balance the entire set of coefficients and derivatives are real artists.

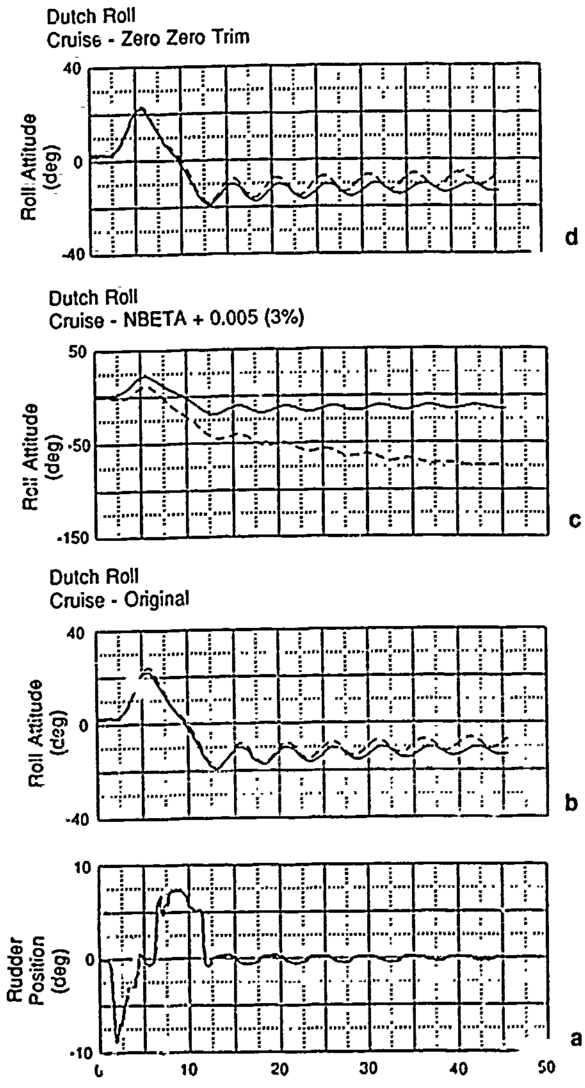


FIGURE 13. Effects of Rate Mistrims and Derivative Error

An example of the sensitivities to subtle and minor changes in trim and stability derivatives are illustrated in figure 13 which show the response of the aircraft bank angle of a C-141 airplane to a rudder doublet (Figure 13a) that excites the dutch roll mode of the airplane. With the simulator trimmed so as to account for the small angular rates due to mistrims of the test data (less than .1 degrees per second) the first cut using the flight test derived derivative set for this flight condition gives a reasonably good match as seen in 13b. But if the angular rates are taken to be zero at the trim condition, the simulated bank angle is seen to diverge as seen in Figure 13c, and exceeds fifty (50) degrees error by the end of the maneuver. It is very important to begin matching maneuvers from proper trim conditions.

The effect of simulator derivative errors in C_{l_β} of 3 percent, illustrated in Figure 13d, is less dramatic than errors in trim. However, there are noticeable effects on the frequency and damping of the oscillation for this very small error in the value of just one derivative. These kinds of errors can be expected in any of the coefficients and derivatives that affect a maneuver. The result could be very dramatic if the errors interacted adversely in the simulator model.

SIMULATOR ACCEPTABILITY

In spite of all of the good hardware, software and data, simulators still will not fly like the airplane. This fact is true because certain flight conditions and types of maneuvers can never be properly represented by the motion or visual systems. The reason is that we don't understand all we know about the system we are analyzing.

Parameter identification is not the whole answer to simulator model development. It does a wonderful job of developing the derivatives for large disturbance maneuvers but those modes, and associated derivatives that are not or cannot be excited are not well identified.

Pilot acceptability of a simulation is greatly influenced by the very small vibrations and responses about the trim condition, either level flight or while maneuvering. These responses are impossible for the manufacturer to quantify in the design and development stage and they are very difficult to evaluate in flight but they are important to pilot opinion. First, the forces are very small and the control movements are almost imperceptible. Being small they are in a region that is most likely non linear so that the traditional equations of motion do not apply. Analysis of this particular area deserves closer examination both in flight and in the simulator.

Even though there has been much progress, we still need to focus on the common complaints of experienced and knowledgeable pilots that still remain even for the best and most sophisticated of simulators. The more common ones are outlined in Table I but surely there are others. The ones enumerated are clearly related to the data issues that are the subject of this paper.

TABLE I

COMMON COMPLAINTS FOR THE BEST OF SIMS

General Comments:

- Simulators Can Pass All Objective Match Tests and Still Not Fully Represent the Aircraft – Small Disturbance Effects Are the Key Reasons.
- Simulators lack the crisp response and feel of the real aircraft.
- Large disturbance motions are generally satisfactory but small inputs and response are where the pilot is most sensitive and most critical

Specific Comments:

- Response of Visual System and Instruments to Control Input lags the input (Latency)
- Subtle vibrations and sounds are not represented properly
- Ground Handling Motion Peculiarities are not correct
- Visual Peripheral Cues are limited
- Low Daylight brightness especially internal to cockpit
- Highly Maneuverable aircraft can't simulate the steady state g-loads
- Ground Effect of the simulator is usually lower where lift and, to a lesser extent, pitching Moment is concerned.

"Full flight simulators such as those that meet FAA levels C and D are very good at creating an illusion of flying the 'real thing' and can make the pilot sweat. But the illusion is shut down and it becomes just a trainer when minor distractions occur" (e.g. an atypical response, the flickering of a light, the extraneous noise of a fire bell in the simulator next door, a comment or even worse a freeze of the simulator action by the instructor, etc.)

Atypical responses are an engineering problem but many of the other distractions are a training procedures problem. Perhaps it would be better to temporarily save a particular training exercise and return to it later at the critical point to critique or re-initiate it.

CONCLUSIONS

No matter how good the hardware and the software, the final simulator will still be a poor simulator if the type and quality of the data is not good.

The traditional approach to generating simulator design data from manufacturers design data is lacking in that it leaves many holes in the data set where parameters must then be adjusted and filled by highly skilled "artists" to match the specific flight condition and also to make smooth transitions from one flight condition to another in a seamless manner. This procedure is time consuming and introduces unreal characteristics into the data model.

The flight test approach provides a more consistent and coherent data model that represents more of the variables as we understand them today. Flight test generated models tend to be insensitive to errors in weight, c.g 's and inertias because the same model is used to quantify the model in flight as is used in the simulator on the ground. The model derivatives obtained in flight contain all of the flexibility characteristic of the airplane without having to overtly know what modes and stiffnesses of the airplane really exist. Furthermore, if the validation data is obtained on the same airplane as the design model data, the two sets of data are more coherent because the variables that might otherwise be introduced by discrepancies between two different data acquisition systems are not present.

While flight test is the preferred way to obtain both design and validation data it is not the whole answer, especially for small disturbances where nonlinear effects of control friction and small disturbance aerodynamics are present. A considerable amount of study is needed to investigate, quantify and model these effects. Until more knowledge is gained in this area, final adjustment to the simulation model will have to be based on qualitative comments of pilots and the skill of simulator engineering/artists.

Finally, experienced flight test people grossly underestimate the quality and care that must go into producing flight test data for simulator validation let alone the gathering of design data. In large flight test organizations it can be very difficult, if not impossible, to get everyone sensitized and religiously committed to attending to every detail in order to squeeze every last bit of performance out of the data system.

The demand for higher quality training devices and greater training credits will continue to push the simulation industry for better hardware and software but we must remember that the simulator can be no better than the data and the data model mechanization.

In the final analysis, simulator acceptability is totally determined by the pilot. But objective standards have gone a long way toward providing a product that is very close to being acceptable. Increased knowledge and understanding of the secondary aerodynamic derivatives and nonlinear characteristics of the airplane and control system could bring the objective and subjective criteria for acceptability in much closer agreement.

REFERENCES:

1. Hess, Robert A; Hildreth, Bruce L.; Numerical Identification and Estimation of an Efficient Method for Improving Simulation Fidelity I/ITSC Proceedings Nov. 1990 pg 123.

2. Kohlman, David L.; A Comparison of Flight Test and Predictive Results from a Series of Simulator Data Generation Programs, AIAA 88-2131

3. Humphrey, Tom; Winters, Mark; Definition and Validation of the Flying Qualities and Performance Test Criteria for the Modern Operational Flight Trainer; I/ITSC Proceeding, Nov. 1990, pg 130.

4. Tafts, William D.; Kraft, James N.; Boeing Simulator Approval Process, AIAA-85-1736.

5. Jacobs, Robert S.; Roscoe, Stanley N.; Simulator Cockpit Motion and the Transfer of Flight Training, ISBN 0-8138-1925-3

UTILIZING A BLADE ELEMENT MODEL FOR HELICOPTER PILOT TRAINING

Martin T. Jakub, Leonard Richmond, Allen Tracy
Eyring, Inc.
Salt Lake City, Utah

ABSTRACT

The simulation of rotorcraft flight is difficult. One especially difficult challenge is to model the aerodynamics in the nonuniform wind environment near the ground. The rotor map or rotor disc aerodynamic models must be replaced by a more sophisticated blade element model, and there needs to be a way of modelling the complex airflow around solid objects. This paper shows a model of airflow around ship superstructures, which is important in pilot training for shipboard operations. We show a method of describing the velocity field in the vicinity of the rotors, suitable for communication to dedicated blade element computer systems.

INTRODUCTION

The "holy grail" of simulation remains nap-of-the-earth helicopter flight. This combines the difficult aerodynamics of helicopters with high visual system demands due to rapidly changing scene content. Part of this challenge is modelling the effects of localized winds near the ground, and in particular the wind environment near ships.

In the mid to late '60s, research was done into carrier airwakes, eventually resulting in the carrier landing disturbance model described in MIL-F-8785C and MIL-STD-1797. This research was designed to understand airflows that were complicating fixed-wing aircraft carrier landing, particularly the burble or "rooster tail" that occurs just before the plane crosses the deck. Eventually the pilot-training simulators for fixed-wing aircraft incorporated models based on the studies and/or the mil standards[1,2]. For rotary aircraft, however, a more significant piloting difficulty results from wind patterns caused by the ship superstructure. Furthermore, helicopters land on many types of ships, not just carriers.

The safety envelope for shipboard helicopter operations has been explored by the Naval Air Test Center. Given a specific ship/aircraft combination, the ability of the pilot to perform various tasks is evaluated. The pilot faces the cumulative obstacles of ship motion, ship airwake turbulence, obstructions to field of view or landing aids, and wind-over-deck factors. Such evaluations, called "dynamic interface studies," are expensive, and there are many ship/aircraft combinations to be evaluated. Could computer simulation be a cost-effective substitute for field tests? Unfortunately, there is not enough quantitative experimental data with which to develop models. If models could be developed and tested against adequate experimental data, such models could be used to estimate new ship/aircraft combinations.

That time is not yet here. Presently, discussion continues among the experts in the simulation and fleet operations communities toward that end[3,4,5].

During the last 30 years, remarkable progress in the computer industry has made it possible to integrate lift and drag along a rotary wing in real-time. Such a model is called a blade element model. Until recently, such models have been used only in non-real time applications to design rotary aircraft and to design rotor maps (lookup tables) to be used in real-time flight simulation. Now it is possible to incorporate the blade element model directly in the real-time simulation.

Naval Training Systems Center has tasked Eyring, Inc. with upgrading the pilot-training simulators for the CH46 tandem rotor helicopters, and specifically to utilize a blade element model and to improve the simulation of the localized wind near ships and formation aircraft. The project is an effort by the government to improve simulation in light of the known difficulties of the task[6].

BLADE ELEMENT MODEL

To provide needed computing resources to add blade element modeling to the existing simulator, we have added a "compute box" dedicated to that task. It is connected to the main simulation computer via a parallel interface. All other simulation tasks (including fuselage aerodynamics) remain in the main simulation computer.

Using a separate compute box introduces several new concerns. A transport delay is one inevitable result, since data must be passed to the box on one iteration and returned on the following iteration. Our main simulation runs at a relatively high iteration rate of 64 Hz, so the additional delay is only 1/64th of a second. Another concern is that of frequency folding. Since the main simulation effectively "samples" the output

of the compute box, it is possible that high-frequency components of the aeroforces could be folded into lower frequencies. When this difficulty was studied at NASA Ames, it was found that these problems can be largely alleviated by running the blade element model at an even higher frequency, and by using filters judiciously[7]. The blade element model runs at three times 64 Hz, which prevents aliasing of all harmonics through the 7th of the 6-per-revolution beat experienced by the tandem 3-blade rotor hubs of the CH46, thus minimizing the problem.

The performance specifications for the blade element model are for an azimuth step of 8.25 degrees and 10 elements per blade, 3 blades per hub, and 2 hubs per helicopter. To meet this requirement, the compute box is designed using a VME bus chassis with digital signal processor (DSP) cards. DSPs are a fairly recent development in chip technology, specifically designed to perform signal processing. What makes DSPs ideal in a compute box for integration is that a single chip can perform a floating point addition and multiplication concurrently in a single cycle. The compute box contains a total of six DSPs. Maximum peak performance is 192 million floating point operations per second, roughly 200 times the speed of a VAX 11/780.

The inputs to the blade element model are passed to the main DSP for each of the two hubs via a 68030 processor, which handles communications between the individual DSPs and between the compute box and the main simulation computer. Calculations general to a hub are completed and results are passed to subservient DSPs, which then perform the calculations for the other blades. These results are passed back to the main DSPs to be summed. The aft hub calculations are passed to the forward hub's main DSP to be included in the total force and moment results and passed back to the host via the 68030. Since most of the time is spent performing the individual blade calculations, the six DSPs form a very efficient parallel computer system for blade element calculations.

The software for the blade element model is written in ADA and is based on the GenHel blade element program[8]. The GenHel program was developed by Sikorsky in the 1970s and has been actively used by NASA Ames for studies of rotorcraft[9]. In short, the GenHel model has been evolving and improving for a number of years, and constitutes a solid starting point for further development. The software model was converted from Fortran to ADA, modified for the rotor hub geometry of the CH46, and enhanced to account for dynamic stall and perturbed airflow effects. Static tests from wind tunnels do not accurately predict the stall conditions or even the lift under non stall conditions for a wing that is cyclicly changing in pitch, since it takes a certain amount of time for the flow to become the more turbulent flow associated with a stall condition or to settle to the laminar flow associated with a static condition. The stall effect is referred to as "lift overshoot" and is important for simulating flight that is at the edge of the speed envelope[10], whereas the non-stall lift correction is necessary even for normal flight. Other modifications were necessary to allow thrust and drag gains for power settling and ground effect conditions, and to simulate local wind velocities near the hub.

THE WIND TO BLADE ELEMENT INTERFACE

The parallel interface to the blade element compute box is of limited bandwidth, so a short and succinct list of input and output variables is a necessity. To describe the localized winds near a ship, the entire wind velocity field must be passed. One way to do this is by approximating the velocity field in a linear fashion; i.e., use the first order multi-dimensional Taylor series approximation:

$$V(x+dx) = V(x) + V'(x)*dx \qquad (1)$$

where:

$V = (u,v,w)$ is a vector function of position, and
V' = Jacobian matrix defined by

$$\begin{matrix} \nabla u \\ \nabla v \\ \nabla w \end{matrix}$$

i.e., the matrix whose rows are the gradients of the components of V.

Although the first order approximation is very simple, it allows for surprisingly complex flow fields. For example, the Jacobian matrix:

$$\begin{matrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{matrix}$$

describes a circular flow around the x axis. A circular or elliptical flow is especially applicable to modelling the standing and trailing vortices that can occur around the ship superstructure, especially near a hangar.

The main simulation computer calculates the Jacobian matrix at each rotor hub and the velocity at the rotor hubs. From this information, the blade element model processors can estimate the velocity field at any region of the rotor disk, using equation (1). The Jacobian itself is estimated by putting an imaginary "pizza box" around the nominal region of the rotor disk, and then evaluating the wind velocities at the centers of the faces of the box. The differences in velocities on opposite faces, divided by the distance between the faces, gives an estimate of the partial derivatives (one column in the Jacobian matrix). For the two Jacobians and hub velocities, a total of 20 floating

point words describes the wind velocity fields in the rotor regions. This is a small enough interface to meet the bandwidth requirements.

The Jacobian matrix is used to calculate the localized wind effects on the fuselage. The yaw component of the aerodynamic torque on the fuselage is

$$N = q^*C_N - K^*(r + (\nabla \times V)_z) \qquad (2)$$

where:

q = dynamic air pressure
C_N = yaw coefficient from wind tunnel data
K = damping factor
r = turn rate
$(\nabla \times V)_z$ =z component of the curl of the wind velocity

The curl of a vector field is a linear sum of the non-diagonal elements of the Jacobian, and can be thought of as the rotational component of the flow. The K*r term represents the damping that occurs when the fuselage is rotating in wind with no rotational components. Similarly, a nonrotating fuselage with the wind rotating about the aircraft will also result in a torque. Consider an example situation where this would arise. Imagine taking off from a ship that is facing into the wind. The helo pad is directly downwind of a hangar that acts as a wind shield. If the helo is hovered to a position where the nose is exposed to the wind but the aft section is still in the wind shadow, a torque will occur and the helo will turn.

WINDS

The discussion of winds in this paper will be limited to the ship-wind interface, formation aircraft, and ground effects.

Ship-Wind Interface

Modelling of the ship-wind interface can be divided into two basic areas: turbulence and the ship-airwake structure[17].

Turbulence is so complicated that at present it cannot be computed in real time. However, by studying the statistical properties of turbulence it is possible to build functions that pilots find similar to the real thing[19]. By studying the energy spectrum of the atmosphere, two widely accepted statistical models have been developed: the Dryden and Von Karman models. These models construct an energy spectrum based upon the correlations of spatial wavelengths[14]. The Von Karman forms are considered to be the more accurate of the two, so they became the starting point for our model. The Von Karman spectrum describes atmospheric turbulence, but does not represent the interaction of turbulence with a ship at sea. Due t the low altitude and interaction with the ship, th equations need to be altered.

Dr. Val Healy, of the Naval Postgraduate School, ha revised the Von Karman equations to include thes considerations[16], resulting in the following longitudin autospectrum:

$$\Phi(\Omega) = \frac{4\sigma^2}{(1 + 70.8\tilde{n}^2)^n} \qquad (4$$

where

$\tilde{n} = \Omega L/U$
U = wind velocity
$n = 5/6$
L = turbulent length scale
σ = the standard deviation or turbulence intensity
Ω = the spatial frequency

From this spectrum, a digital filter is derived with transfer function whose magnitude squared matche this spectrum. This filter is then applied to whit noise to yield a simulated turbulence with th appropriate energy spectrum[19].

Ship-airwake is the second component of the ship wind interface. The wind whipping over the dec produces a steady-state airflow pattern behind th superstructure of the ship. This steady state flov consists of various vortices and a component above th deck in the direction of the wind.

Because this flow pattern is very complex and a adequate experimental data is currently unavailable, a accurate statistical model has been impossible t devise[6].

A previous attempt at a ship-airwake model was mad by Fortenbough using a method called Strouha scaling. Data was gathered from a Boeing win tunnel experiment on a 1/50 scale model of an FF 1052 class destroyer. The data was put into tabl form and then scaled to fit other ships[18]. Thi approach has two problems. First, it assumes tha deck structures of ships are similar, which in mos instances is not the case. Second, wind speeds in th tunnel were measured at points too far apart to get a accurate idea of flow pattern, especially that o vortices. Healy states that the method is very crud and that "results can be expected to be as accurate a picking random numbers"[16].

An alternate method is to pattern the airflow aroun the ship superstructure from the study of aerodynamics of buildings. Structures or groups of structures o ship decks can be characterized as bluff bodies, whic are described as short, squat buildings. From this an studies of helium bubble patterns behind models i wind tunnel tests a general description of the airflo

can be devised[15].

With wind heading at 0° relative to the ship, a standing vortex forms directly behind the hangar (figure 1). A cross-section of this vortex is approximately circular with a diameter equal to the height of the hangar. This vortex extends clear across the face of the hangar. There are also two trailing vortices which start along the sides of the hangar and flow back along the side of the ship until they die out. Behind the standing vortex and between the trailing vortices, the pattern dives down towards the deck, flows back, and rises a distance aft of the ship.

When the wind moves off the 0° heading, this vortex pattern is somewhat altered. For instance, if the wind blows from 330°, as in figure 2, the standing vortex behind the hangar will begin to shorten on the left side and angle behind the superstructure.

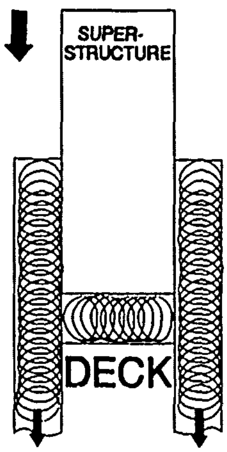


Figure 1. Wind from 0° - Vortices

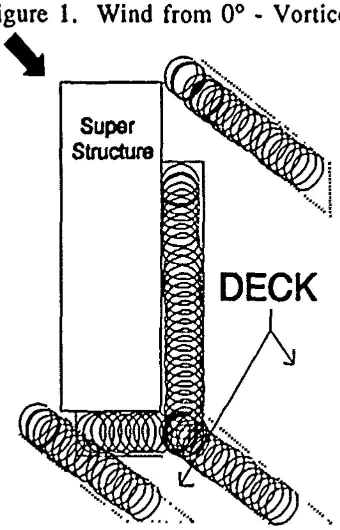


Figure 2. Wind from 330° - Vortices

At the same time a standing vortex forms on the right side of the hangar, beginning at the aft corner and growing toward the front corner as the wind angle decreases towards 270°. The trailing vortices now flow downwind from the front right and aft left corners of the hangar. In addition, a third trailing vortex forms where the two standing vortices meet at the right aft corner. This vortex is weaker than the other two. The "deck" component of the airflow shows the same general behavior as for 0° heading wind as it flows downwind. The vortices line up with the relative wind as its angle changes.

An analytical model of the ship airwake has been designed, based upon this flow pattern and the previously mentioned Boeing data[13]. The Boeing data is used as "starting points." In other words, with a mean wind speed of 20 feet/second, the data may show an x-component velocity of 6 feet/second at a point on the perimeter of the standing vortex. This is used as the starting velocity, and the conservation of momentum is used to calculate x-component velocity as we move further inside the vortex. Again, if a certain velocity is measured at a point just aft of the vortex, it is used as a starting point and is then manipulated mathematically in accordance with the flow pattern.

This analytical model is dependent on the factors of mean wind speed, wind heading in relation to ship heading, and the dimensions of the ship and its superstructure. Therefore, it can be applied to any ship by changing only those inputs contingent on the dimensions of the specific ships.

Combining the ship-wind elements: As mentioned previously, to estimate the Jacobian matrix of the wind velocity at the hubs, it is necessary to calculate the wind velocities in the rotor sampling box shown in figure 3.

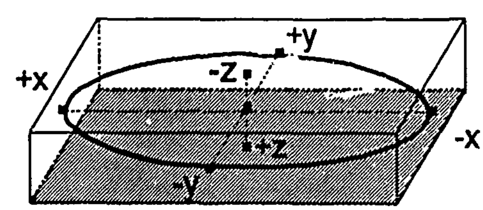


Figure 3. Rotor Sampling "Box"

For each point, a gain value is calculated to indicate if it is in the region of a particular wind component (eight vortices and the deck component). If a value greater than 0 is computed, velocities are then calculated for that wind component.

Three arrays of size 3x14 (x,y,z component velocities for each point) are used to store velocities for each of the three wind component types (standing vortices, trailing vortices, and deck component). These are then added together along with turbulence velocities to form one 3x14 array.

Formation Aircraft Disturbance

This module simulates the disturbances encountered by a helicopter flying in formation behind another helicopter. The formation aircraft, which is also a CH46 tandem rotor helicopter, is modeled as a disc whose radius is the distance from the center of gravity to the rotor tip. Flight instructors have determined that the disturbance envelope may be represented as a cylinder (figure 4) which trails behind and below the lead aircraft at the skew wake angle.

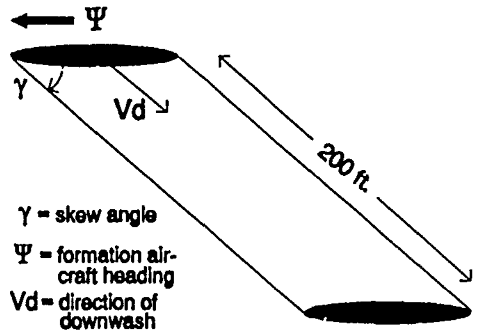


Figure 4. Formation Aircraft Disturbance Envelope

Pilots have stated that the wash tends to roll trailing aircraft in towards the center of this cylinder as in figure 5.

This effect is because of wing-tip vortices shed from the lead aircraft. This effect is to be simulated by constructing two vortices, one on either side of the radial axis. The magnitude of the vortex velocities will dissipate down the cylinder length and will completely disappear at a distance of 200 feet. In addition, the wash produces a strong wind component in the direction of the cylinder axis which dissipates at 200 feet down the axis and radially away from the center. Thus, rolling effects will become more severe towards the center of the cylinder. All effects are washed to mean wind values at a distance of 2 cylinder radii from the center.

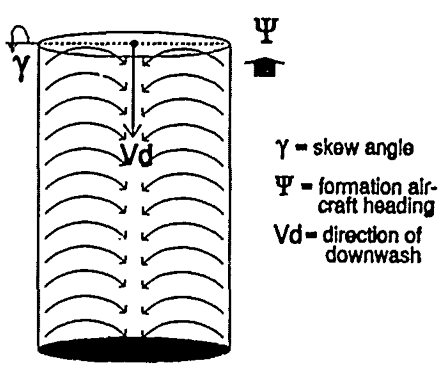


Figure 5. Formation Aircraft Disturbance Pattern

The location of each point is computed in the coordinate system of the trailing cylinder. The cylinder is then tilted to form a right cylinder to more easily facilitate construction of the vortices. Velocities are initially calculated in vortex coordinates and are then converted back to cylinder coordinates and finally to earth coordinates.

In addition to these steady state effects, three-dimensional turbulence is added in proportion to the component velocities already calculated.

Two velocity arrays and two Jacobian matrices are computed in the same way as those in the ship airwake routine.

Ground Effects

When the helicopter is within a rotor diameter of the ground, ground effects from the rotor downwash can significantly decrease required torque. Existing models calculate "in hover" ground effects, ignoring the fact that lateral movement decreases ground effects.

A "mirror" model has been devised to address this problem (figure 6). In this model, the downwash reflects off the ground, and the reflected wash results in increased lift. For a transition from hover to forward flight, this model correctly predicts that the helicopter will "roll off the cushion" of ground effect as the aircraft pitches forward.

The model starts by modelling each rotor as a separate disc. Downwash velocities are passed from the Blade Element Model and a cylinder is constructed for each disc in much the same way as in the Formation Aircraft Module. The cylinders extend to the ground and are reflected back up. If the aircraft is in hover, the cylinders reflect straight up, and will result in full ground effects.

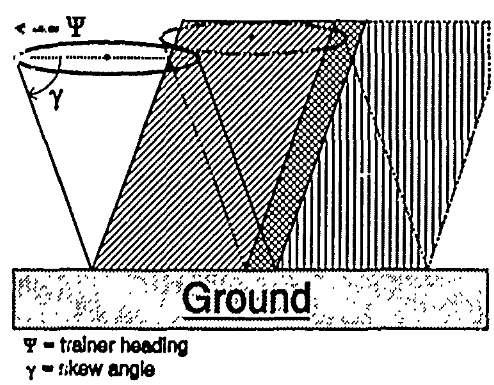


Figure 6. Ground Effect Mirror Model

If there is a forward velocity, the cylinders will reflect backwards resulting in less effect on the forward hub.

The aft hub will experience increased effects from the forward cylinder and decreased effects from the aft cylinder.

The cylinders are analyzed to check the amount of area which intersects with the rotor discs. Four gains are calculated based upon the amount of intersecting area. These gains are multiplied by the hover thrust gains to
produce a total thrust gain.

Calculating the Jacobians and Velocities

Finally, the 3x14 arrays from both the Ship Airwake module and Formation Aircraft Disturbance module are added together. The sample points at the hub locations are put into two velocity arrays, one for the forward hub and one for the aft hub. Component velocities at the remaining points are used to form the Jacobian matrices for each hub. The two velocity arrays and two Jacobian matrices are passed to the Blade Element Model for application.

CONCLUSIONS AND FUTURE DEVELOPMENTS

The project described in this paper is currently being implemented. Soon pilots and instructors will be able to provide their opinions as to the accuracy and value of the ship operations simulation for windy conditions. There is also considerable interest as to how the fidelity of a blade-element based simulation will compare to the rotor-map model presently employed. We have high hopes that a blade-element model will provide a better global model of flight, especially for the edge-of-the-envelope conditions that are not considered in the design of rotor maps.

For the immediate future, localized wind will be simulated by using functions. Table-based lookups are memory-intensive and rely on vast amounts of data collected for each specific ship. The solution of fluid dynamics equations in real-time is at present science fiction and is not a viable alternative. At some future point, this fiction will become fact. At that time, it will make sense to use the visual data base to solve for wind flow patterns that match the constraints of the terrain and objects in the database.

Meanwhile, we can hope that enough quantitative data will be collected to provide a better understanding of airflow around ships. With such data at hand, the task will be to improve the simulation until it can be used to predict or extrapolate the difficulty of shipboard operations by those doing dynamic interface studies.

REFERENCES

1. Weir, D.H., "Analysis and Comparison of Carrier Airwakes," Systems Technology, Inc., STI Technical Report No. 158-1, April 1966.

2. Durand, T.S., "Carrier Landing Analysis," Systems Technology, Inc., STI Technical Report No. 137-2, February 1967.

3. Carico, D., McCallum, K.E., Higman, J., "Dynamic Interface Flight Test and Simulation Limitations," Eleventh European Rotorcraft Forum, Paper No. 100, London, England, September 1985.

4. Anon., "Helicopter/Ship Dynamic Interface (DI) Working Group Meeting Proceedings," Naval Air Test Center, April 1988.

5. Anon., "Minutes of the First Aerodynamic Ship Interface Conference," Washington D.C., November 1989.

6. Galloway, R.T., Frey, F.E., Carrico, D., "The Ship Environment Simulation Problem," AIAA paper No. 3171-CP, September 1990.

7. McFarland, R.E., "The N/Rev Phenomenon in Simulating a Blade-Element Rotor System," NASA TM-84344.

8. Howlett, J.J., "UH-60A Black Hawk Engineering Simulation Program," NASA CR-166309, December 1981.

9. Houch, J.A., Moore, F.L, Howlett, J.J., Polecat, K.S., and Browne, M.M., "Rotor System Research Aircraft Simulation Mathematical Model," NASA TM-87629, 1977.

10. Prouty, R.W., <u>Helicopter Performance, Stability, and Control</u>, pp. 220-225, PWS Engineering, 1986.

11. Campbell, Warren C. "A Spatial Model of Wind Shear and Turbulence for Flight Simulation," NASA Technical Paper No. 2313, May 1984.

12. Cermak, J., "Aerodynamics of Buildings," Annual Review of Fluid Mechanics, Vol. 8 (1976), pp. 75-106.

13. Garnett, T.S., "Investigation to Study Aerodynamic Ship Wake Turbulence Generated by an FF1052 Frigate," Boeing Vertol Report D210-11140-1, Dec. 1976.

14. <u>Handbook of Turbulence, Volume 1</u>. Frost, Walter and Moulden, Trevor H., editors. Plenum Press. NY.

15. Healy, J. Val, Ph.D., & Johns, LCDR, Michael, "The Airwake of a DD-963 Class Destroyer", Naval

Engineers Journal, May 1989, pp. 36-42.

16. Healy J. Val. "The Prospects for Simulating the Helicopter/Ship Interface", Naval Engineers Journal, March 1987, pp. 45-60.

17. Military Specification. "Flying Qualities of Piloted Airplanes," MIL-F-8785C, Nov. 5, 1980.

18. Fortenbaugh, Robert A., "Progress in Mathematical Modeling of the Aircraft Operational Environment of DD 963 Class Ships," AIAA Atmospheric Flight Mechanics Conference, Boulder, CO, Aug. 6-8, 1979.

19. Robinson, Paul A., "Modelling Atmospheric Effects for Flight Simulators," AIAA flight simulations conference paper 89-3284-CP, August 1989.

ABOUT THE AUTHORS

Marty Jakub is a scientific programmer at Eyring, Inc. He is the team leader on the project to implement the blade element model on the 2F117 simulator. He holds a B.A. in mathematics from the University of California at San Diego.

Leonard Richmond is a scientific programmer at Eyring, Inc. He is responsible for the blade element software development. He holds a B.S. in physics and mathematics from Louisiana State University and an M.E. in computer science from the University of Utah.

Allen Tracy is a scientific programmer at Eyring, Inc. He designed and coded the winds simulation. He holds a B.S.E.E. from Brigham Young University.

The Challenges of Simulating a Hovercraft Ocean Environment

Mark E. Donner
Hughes Training, Inc.
Herndon, Virginia

Abstract

A critical part of the development of a hovercraft simulator is accurately representing an ocean environment. Unlike traditional aircraft simulators, the dynamics of a hovercraft are driven by the forces produced by the sea medium in an ocean environment. The importance of correctly depicting such ocean entities as a sea state, a plunging surf, and a support ship wake becomes evident when one considers that the methods used in modeling these environments directly affect the judgment and actions of the crew. Not only must the details of these ocean entities be readily identifiable, but their dynamics must accurately represent the real world as perceived by the crew. This was accomplished on the Landing Craft, Air Cushion Full Mission Trainer (LCAC FMT) which has the capability of displaying realistic ocean environments. The objective of this paper is to present how sea states 0 to 4, a dynamic surf, and a support ship dynamic wake were modeled and how some of the limitations of these models were overcome. By enhancing the characteristics of these models and developing creative methods of implementation, Hughes Training, Incorporated, is meeting the Navy's need to provide a realistic ocean training environment. This paper discusses some of the design considerations required to provide a real-world ocean environment as perceived by the users of the Landing Craft, Air Cushion Full Mission Trainer.

The Role of the LCAC in Amphibious Operations

From the time that Julius Caesar developed specialized ships for the invasion of Britain (55 B.C.) to the highly mechanized assault of Inchon, Korea (1950), history has documented the triumphs of successful amphibious assaults.[12] In all cases, these successful operations were engineered by well-trained forces who were able to rapidly expedite their armament ashore. Amphibious assaults are typically planned to stun an opponent. The psychological impact of being vulnerable from a seaward front is devastating to the morale of an enemy.[3] Such maneuvers require specialized equipment as well as a trained fighting force.

Today, the U. S. Navy is adopting an Over-The-Horizon philosophy of ship-to-shore assaults. The premise of this operation lies in the ability of air and sea assault vehicles to depart an Amphibious Task Force and converge on any number of predetermined beaches from a stand-off distance.[8,9] The Navy uses many types of assault weapons to perform such operations including: helicopters (CH-53E, CH-46E, AH-1W), aircraft (Harrier, and perhaps the V-22), personnel carriers (LCM, LCU, LCVP), and a high-speed hovercraft known as the Landing Craft, Air Cushion (LCAC). The speed and range of these weapon systems force an enemy to defend an entire coast until the specific location of the attack can be determined.[11] The mission of the Amphibious Task Force may be merely to present an attack threat, thereby keeping the enemy in a defensive posture, as was the case during the Persian Gulf War (1991).[10,11]

After an initial beach assault, a crucial mission of the Amphibious Task Force is to make preparations for supplying offensive troops. This logistical problem requires vehicles that can disembark from the Amphibious Task Force, transport equipment an appreciable distance over sea, travel inland before unloading, and return to the sea for successive loads. The Landing Craft, Air Cushion is a hovercraft designed to transport various resources and personnel from the well deck of a support ship to a landing zone up to one mile inland.[1,2] Capable of being launched from up to five different support ships, the LCAC can carry in excess of 65 tons of cargo (the approximate weight of an M1A1-Main Battle Tank) at 40 knots during calm seas. The hovercraft literally rides on a cushion of air and is less

susceptible to mines and water obstructions that can defeat other traditional amphibious landing crafts. The flexible air cushion system of the LCAC (Figure 1) allows this vehicle to operate effectively in sea states 0 to 3 and traverse many different types of terrain, including swamps, mud flats, rocks, and various beach gradients. As such, tidal conditions do not affect the operation of the LCAC. The range and speed of this amphibious hovercraft have expanded the potential of amphibious assaults and have contributed to the maneuver-warfare philosophy inherent in the Over-The-Horizon doctrine.

Landing Craft Air Cushion Full Mission Trainer

Hughes Training, Incorporated (HTI) is providing the first Landing Craft, Air Cushion Full Mission Trainer for the Naval Assault Craft Unit 5 (ACU5) at the Naval Amphibious School in Coronado, California. This six degree-of-freedom trainer contains complete Group Commander, Operator, Engineer, and Navigator suites. The trainer accurately simulates the performance, maneuvering capabilities, and craft systems of the actual hovercraft. A Hughes CT6 Image Generator with a Wide II display provides an aggregate field of view of 170 degrees horizontal by 40 degrees vertical. Images are displayed at a rate of 50 hertz on a Mylar mirror from four channels each with a horizontal field of view of approximately 45 degrees (allowing for some overlap at the channel boundaries). This visual display system provides a realistic scene of such operational environments as calm to rough seas, a plunging surf at predetermined beaches, numerous types of terrain, rivers, and inner waterways. Other visual cues include military vessels (flotillas, support ships, and landing crafts), civilian vessels (pleasure crafts, oil tankers, and barges), designated landing zones (identified by beach markers, flares, and a beach master), and civilian population centers (cities, houses, and farms). Not only must the FMT trainee be able to operate among these obstacles, but the instructor has the capability to induce craft system malfunctions as well as create meteorological conditions.

Unlike aircraft simulators, the LCAC is always located relatively close to the operating surface (the crew rides approximately 18 to 23 feet above the surface). The LCAC operator commonly uses the undulation of the terrain or sea surface to maneuver the craft. For example, the craft operator may use the slope of a hill or the swell of an oncoming wave to reduce the craft sideslip; allowing the aft of the LCAC to encounter the rise effectively straightens out the track of the hovercraft. Of course, the LCAC crew was familiar with the visual cues required for training and approached the design of the ocean gaming area with a preconception that these cues would be pervasive in nature and easy to spot. They felt that emphasizing these cues would not produce false training results. Because of the close proximity of the operator to the terrain, and the familiarity of the crew with different types of ocean attributes, a realistic simulation of the LCAC's operational environment posed a technical challenge not encountered before in the training industry. The remainder of this paper will present the design considerations and the lessons learned in making sea states 0 to 4, a dynamic surf, and a support ship dynamic wake successful visual cues for providing Assault Craft Units 4 and 5 with the correct characteristics for effective hovercraft training.

Training in Open Ocean Environments

Accurately predicting the interaction between the craft and the sea is of paramount importance because the LCAC tends to conform to the contour of the ocean. Without this skill, undesirable hovercraft events may occur, such as plow-in of the LCAC bow with an oncoming wave or slamming the hovercraft superstructure into a trough of a swell. A skilled LCAC operator can anticipate an oncoming wave and adjust the controls to minimize the effects of the sea on the craft. HTI and the Navy spent considerable effort reviewing and fine-tuning the visual and dynamic aspects of these sea state environments. As in many aspects of simulation, the solution of providing realistic sea state cues turned out to be as much of an art as a science.

The elevation profiles of the sea states modeled in the LCAC FMT were derived by a non-real time algorithm, developed by ORI Incorporated, that generates ocean gravity wave elevation spectra. This algorithm calculates sea elevations whose desired characteristics are represented by a

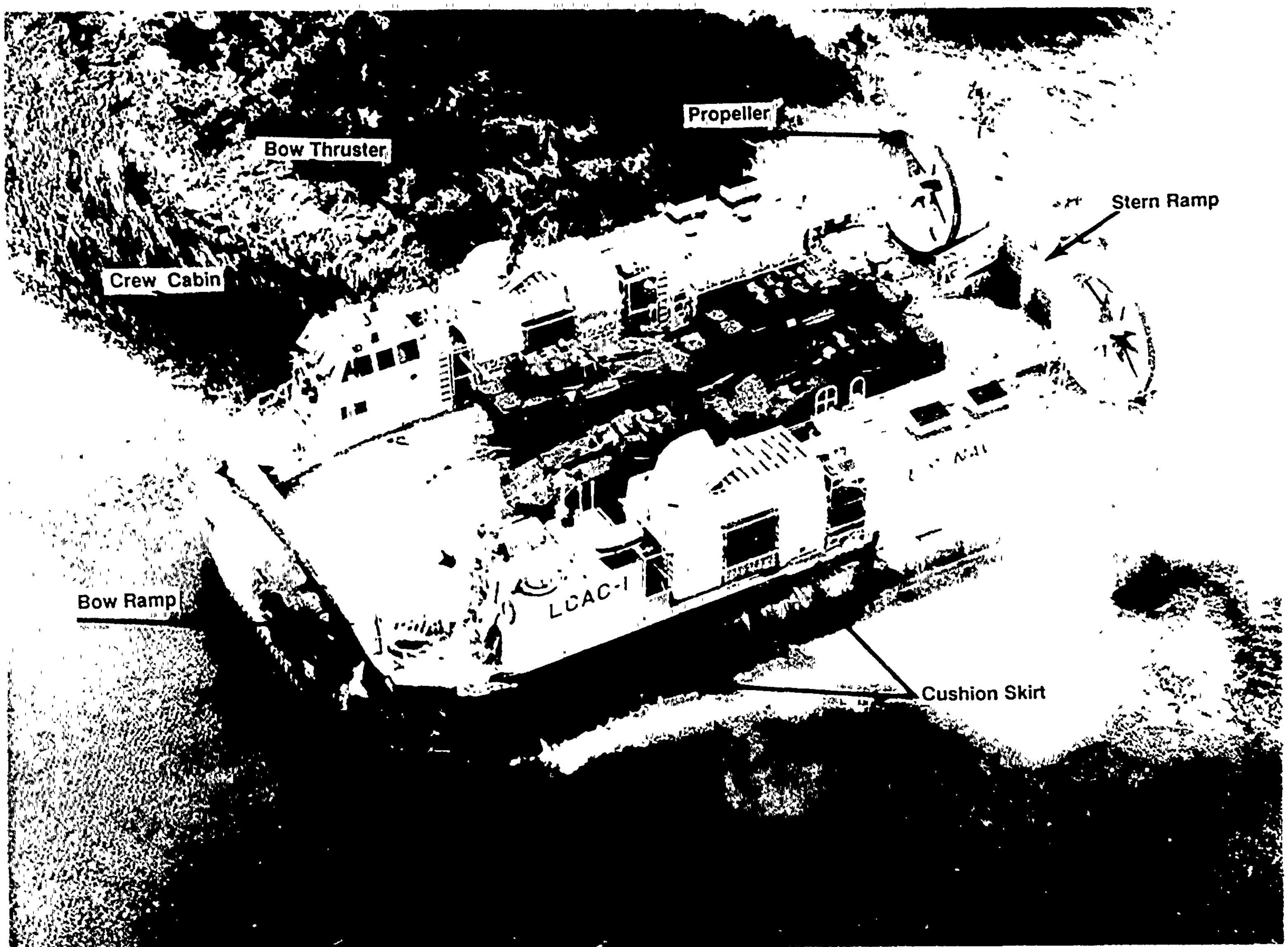


Figure 1 - Landing Craft, Air Cushion The Landing Craft, Air Cushion is an amphibious hovercraft that is propelled by variable pitch propellers and rotatable bow thrusters. The LCAC is suspended above the operating surface by a flexible rubber air cushion system. The craft is used to deliver armament and personnel from support ships at stand-off distances to predetermined landing zones inland.

primary wave. Additional waves are then introduced at varying frequencies, periods, and directions, which consequently alter the characteristics of the primary wave. The resulting wave elevation is expressed by the following Fourier series:

$$H(x,y,t) = \sum_{\ell=-N_\ell}^{N_\ell'} \sum_{m=-N_m}^{N_m'} [A_{\ell m}(\omega_{\ell m})\ COSINE\{(2\pi\ell/\lambda_o)x + (2\pi m/\lambda_o)y - \omega_{\ell m}t + \phi_{\ell m}\}]$$

Where:

$H(x,y,t)$ - The elevation at point (x,y) at time (t), feet.

t - wave period, seconds.

i - wave period indicator.

x - x-position along the wave direction of propagation, feet.

y - y-position along the wave direction of propagation, feet.

N_ℓ' - number of harmonics in x-direction.

N_ℓ - number of harmonics in negative x-direction.

N_m' - number of harmonics in y-direction.

N_m - number of harmonics in negative y-direction.

$A_{\ell m}$ - amplitude for ℓm^{th} wave, feet.

$\omega_{\ell m}$ - frequency for ℓm^{th} wave, radians second^{-1}.

$\phi_{\ell m}$ - random phases for the ℓm^{th} wave, radians.

Sea states 0 to 4 were modeled from these elevation spectra. Each sea state was modeled as a fully developed ocean environment. The gradual build-up of the ocean whenever the instructor changes the sea state or wind velocity was not modeled, the sea direction is driven by the wind heading.

The distortion of the primary wave is controlled by assigning a weight factor to each of the wave components affecting the primary wave, giving them relative importance to the primary wave. The final ocean surface is determined by summing the wave components with respect to the longitudinal and lateral spatial dimensions and time (Figure 2). The distortion of the characteristics of the primary wave was implemented in such a manner as to preserve the directional characteristic of the primary wave. To ensure that the wave spectrum is contiguous at its boundaries, the wave lengths that distort the primary wave length must be integer multiples of the primary wave length.

$$T_j = T_A / j \qquad \text{for } j = 1,2,3,...$$

Such that:

$$\varepsilon_{j\ell} = \text{Minimum } | T_\ell - T'_j | \qquad \text{for } \ell = 1,2,3,...$$

$$\omega_m = 2\pi / T_j', \text{ if wave heading} > \pi$$

$$-2\pi / T_j', \text{ if wave heading} \leq \pi$$

Where:

T_j - allowable wave periods, seconds.

T_A - wave period (9.6 seconds).

T_ℓ - wave period of ℓ^{th} wave, seconds.

$\varepsilon_{j\ell}$ - difference between wave period in question and nearest allowable wave period, seconds.

T'_j - nearest allowable wave period to the wave period corresponding to the gravity wave in question, seconds.

ω_m - frequency corresponding to the adjusted wave period, radians seconds^{-1}.

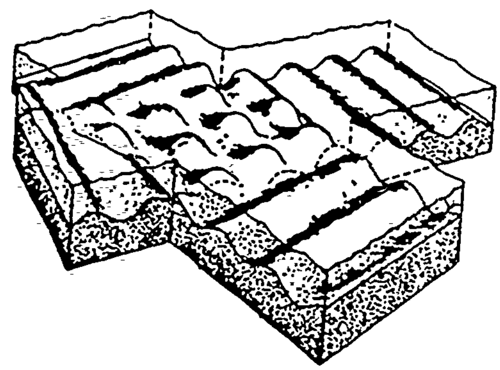


Figure 2 - Wave Interaction Waves propagating from different directions and having unique characteristics are summed to form a resultant sea spectrum. This effect is represented by two such waves at a specific period in time.[6]

The sea state spectra are discretely sampled along their longitudinal and lateral axes with respect to time. This elevation data is then used to generate sets of three-dimensional wave models composed of non-textured polygons whose end points are obtained from the sampled elevation data. The individual wave models are sequentially cycled about the trainee to produce an animated ocean scene. One complete cycle of these models (for a specific sea state) constitutes a complete period of the wave. The specific model currently being displayed in the visual scene is used to synchronize the ocean state vector in the host software with the visual scene.

Sets of the animated sea state models are placed in a 5-by-5 matrix, called a tessellation grid, to give the trainee the perception of a vast ocean. The transition range of the tessellation grid is such that the trainee cannot perceive the boundary of the tessellation grid from the center square. Whenever the LCAC exits the center, the tessellation grid is reconstructed in such a manner as to place the LCAC once again in the center of the 5-by-5 matrix. This method of rearranging the sea states whenever the eye-point exits the center of the tessellation grid is termed "sea-jumping" (Figure 5a). By restructuring the tessellation grid in this manner, the trainee can maneuver throughout the gaming area and perceive an infinite ocean.

The mathematical representation for the sea state spectra can be made as realistic as desired by increasing the number of longitudinal and lateral samples of the resultant wave or by increasing the number of waves that make up the resultant wave. However, the construction of sea state models from the generated elevation spectra is constrained by the number of polygons that can be displayed in real time and the number of models used to represent the sea state. For the CT6 Image Generator, approximately 1500 polygons per channel can be displayed at a time, and the number of models that can be associated with a given sea state is limited to 128. Another factor that affects the creation of the sea scene is the need to preserve the power spectra of the sea state. To obtain reasonable representations of the power spectra of the seas, at least one wave component having a frequency lower than the modal frequency, one wave having a frequency within 10 percent of the modal frequency, and one wave having a frequency higher than the modal frequency were included.* As a result, the wave period required to preserve the power spectra of the seas was determined to be approximately twice that of the modal frequency. These limitations were accounted for in a reduction factor and a power spectra factor used to determine the length of the final wave as follows:

$$\lambda_o = \alpha\beta\,(2\pi g/\omega_m^2)$$

$$K_o = 2\pi/\lambda_o$$

$$T_m = 2\pi/\omega_m$$

* - The modal frequency describes the rate at which the overall sea state spectrum peaks.

Where:

λ_o - *fundamental wavelength, which is also equal to the sea state model length and width, feet.*

g - *gravitational constant (32.174 feet second^{-2})*

α - *reduction factor, 0.83.*

β - *power spectra constant (used to preserve the power spectra of the sea states), 2.*

ω_m - *modal frequency: The frequency at which the resultant sea state spectrum peaks is referred to as the modal frequency, radians second^{-2}.*

K_o - *fundamental wave number, feet^{-1}.*

T_m - *modal period, seconds.*

After the Navy agreed that the ocean scene gave the dynamic cues necessary for training, and the waves were representative of the desired ocean sea states, the wave models were smooth-shaded to give the waves a curved appearance. The smooth shading also eliminated any detection of the individual polygons within a given model of the sea state. The color of the polygons became darker as the polygon normal became horizontal. This effectively lightened the back of the waves and darkened the wave face (the slope of the sea state polygons are more vertical on the front of the waves).

In creating the final oceanic scene, the standard illumination practices of a simulated environment were modified. Diffuse illumination, the illumination that accounts for the reflective energy of the hemisphere making objects brighter irrespective of orientation, was used to simulate the time of day. Direct illumination, the direct energy from the sun that makes objects appear brighter on the side facing the sun, was altered to highlight the specific aspects of the sea state. By keeping the sun angle at a fixed offset from the horizon and setting the heading of the direct illumination to that of the LCAC, the wave contrast is lighter when viewed from the direction of propagation, giving it a froth or white-cap appearance; the waves appear to have more contrast when viewed as oncoming, thereby giving the wave face a more distinguishable dynamic appearance (Figure 3). This, in addition to actual white-caps modeled at higher sea states, enhances the directional characteristics of the ocean and gives the Navy a sea state gaming area that provides the LCAC operator with the optimum realistic sea state training cues.

Some of the difficulty associated with reaching an optimum visual sea scene was in the determination of the wave elevations. Experience shows that an observer on a vessel at sea commonly identifies the significant wave height, the average of the largest one-third of the waves, as the sea state wave elevation.[6] This is due to the difficulty the observer has in distinguishing the movement of the sea from his ownship platform. The sea states were baselined without the advantages of the motion cues of the FMT. HTI, as well as the Navy, had difficulty determining the specific dynamics of the sea state without the motion cues. It should also be noted that the ocean scenes in the LCAC FMT were constructed without benefit of textured polygons. Had textured polygons been used, perhaps the specific attributes needing enhancement would have been more readily identifiable.

Land-to-Sea Transition Training

Most people have gained their experience and knowledge of waves approaching land by watching them from the coast. As waves encounter shallow water, their characteristics begin to change; the wave height increases and wave length decreases. Consequently, as the wave approaches the shore a wave crest curls over a large air pocket and eventually collapses into a smooth splash-up to the shoreline. Under certain circumstances, the energy produced by a breaking wave can cause dramatic damage to the LCAC. To minimize the dynamic effects of the surf on the hovercraft, a craft operator typically examines successive spills of the surf to time the departure from the beach in order to avoid encountering a wave as it begins to break. Such a land-to-sea transition is a skill generally learned through experience, where even well-qualified operators have incurred LCAC damage. Because

Figure 3 - Modeled Sea State 3 The sea states were modeled by continually cycling sets of three-dimensional wave models about the trainee. The waves appear lighter when viewed in the direction of propagation (top), giving a froth or white-cap effect, and have more contrast when viewed as oncoming (bottom), highlighting the wave trough and enhancing the sea state dynamics.

of their frequent occurrence in nature, surf transitions are involved in most land-to-beach operations; consequently, the Navy places a high training value on accurately depicting surf zones in the FMT.

The characteristics of surf zones vary with the tide, sea state, topography under the ocean; as such the analysis of their impact on the LCAC by the operators is subjective. HTI had difficulty identifying the specific attributes of a generic surf which can cause a LCAC Craftmaster to be concerned when attempting to transition through it. At the same time, a surf model so violent that an experienced hovercraft operator would likely elect to abort a mission was avoided. The visual cues of the surf build-up, spill, and splash-up modeled in the FMT had to challenge a Craftmaster and force analysis of the surf prior to performing a land-to-sea transition. Only after reviewing several attempts by HTI at providing a satisfactory representation of a generic surf zone was the Navy satisfied with the model.

The construction of a surf model began with generic digital elevation data that was scaled to represent a sea state 3 open ocean gravity wave evolving into a single 6-foot plunging surf. As with the sea state data, the surf elevation data was used to create sets of individual three-dimensional surf models that were sequentially cycled at specific beaches in the FMT. The surf elevation data was provided at discrete intervals from open ocean to shore with respect to time. This elevation data was used to generate sets of three-dimensional surf models composed of non-textured polygons whose end points were obtained from the sampled elevation data. The individual surf models were sequentially cycled at fixed locations in the FMT to produce an animated plunging surf scene.

Early attempts at modeling the surf used two parallel sets of identical surf profile data separated by the length of the associated sea state (in this case, sea state 3). The surf model was constructed by placing polygons whose end points were defined by coincident points along this profile. This method produced a surf zone that spilled uniformly across the wave face and appeared too mild to cause an experienced LCAC operator to hesitate at a land-to-sea transition. The final model of the surf zone used an additional modified surf profile that was introduced between the previous two digitized profiles. This modified surf profile was created such that it evolved faster than the digitized data profiles. The surf was then constructed by placing polygons whose end points were defined by the digitized data and the enhanced surf profile.

The introduction of the enhanced surf profile increased the number of polygons required to construct the surf model, thereby enhancing the detail of the surf. Additionally, the wave slope (the ratio of the wave height to wave length) was increased. This affected the orientation of the polygons that made up the surf such that normals to the polygons became more horizontal as the wave spilled. As a result, the contrast of the wave face increased, giving it a more violent appearance as the wave began to curl. By temporarily and spatially offsetting the enhanced surf elevation profile, the surf lost its uniform appearance. This provided additional cues that forced the LCAC operators to predict the surf dynamics prior to attempting a land-to-sea transition.

Separate polygons were added to the surf model independent of the profile data to highlight the froth associated with a plunging wave. White polygons were introduced as the surf curled and spilled up to give the surf a more realistic appearance. The surf curl which started as a modified profile began to spill, and spread laterally along the surf face until the entire surf was plunging. The spill-up polygons became transparent as they receded toward the ocean. This gave the effect of the surf froth disappearing into the sand, beach cusps appearing on the shore, and enhancing the terrain cues near land(Figure 4). This also helped the crew to identify the surf spill and splash-up from a seaward perspective (crucial in sea-to-land transitions).

In nature, ocean waves diminish in amplitude and form a wave parallel to the shoreline as they evolve into a surf.[6] To create a natural scene between the open ocean sea state and the surf zone, adjustments to the surf elevation had to be made in the surf profile to account for the spatial and temporal continuity between the animated sea state 3 model and the surf zone. A transition zone was modeled between the open ocean sea state 3 and the area of the surf where the surf actually curls and spills. In this transition zone,

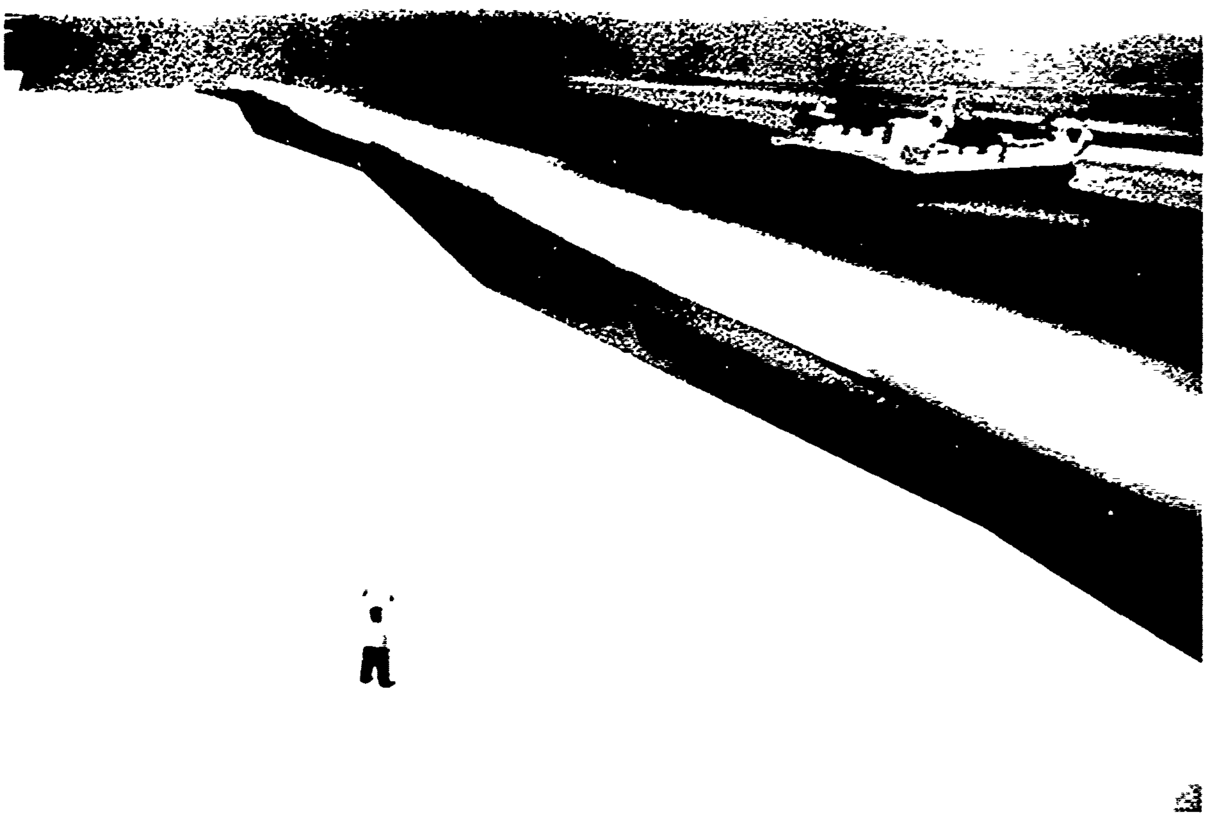


Figure 4 - Modeled 6-Foot Plunging Surf The open ocean sea state 3 gravity wave evolves into a 6-foot plunging surf at predetermined beaches. The dynamics of the surf are highlighted by the addition of froth polygons that make up the surf curl.

surf elevations and open ocean elevations are blended so that the visual continuity between open ocean and the breaking zone was maintained. The elevation of the sea state to surf transition zone was determined as follows:

$$H_T(x,y,t) = W_{oo}\, H(x,y,t) + W_s\, H_s(x,y,t)$$

Where:

$H_T\ (x,y,t)$ - *elevations in the transition zone, feet.*

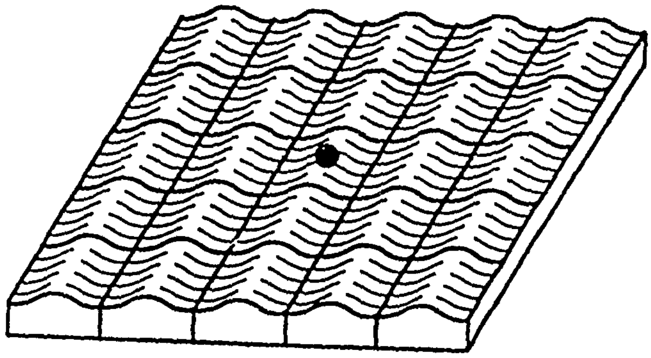
W_{oo} - *open ocean weighting factor.*

W_s - *surf zone weighting factor.*

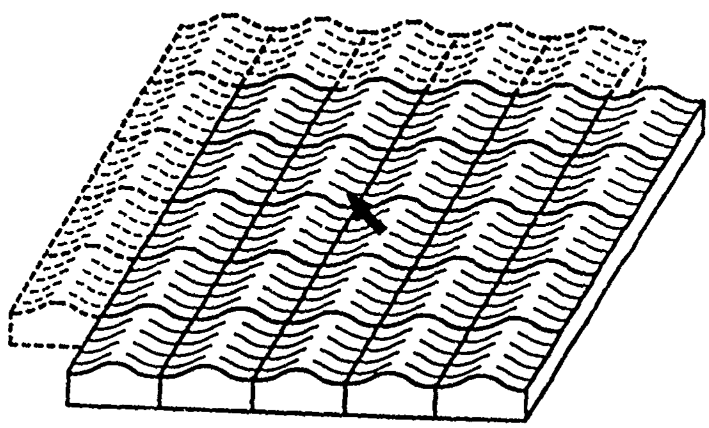
$H(x,y,t)$ *open ocean elevations, feet*

$H_s(x,y,t)$ - *elevations in the surf tessellation area, feet.*

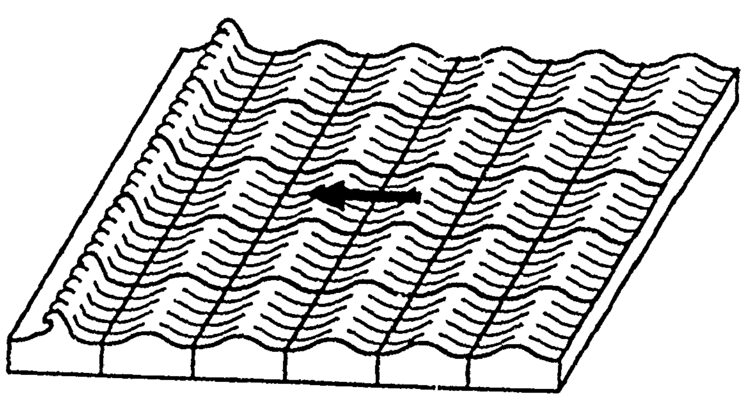
When the LCAC travels within a fixed distance of shoreline where the surf exists, the sea-jumping algorithm associated with the sea state tessellation grid was altered. The sea state was no longer reconfigured in the direction of the surf zone. Whenever the LCAC exits the center grid in any direction other than that of the shoreline, the tessellation grid is reconstructed in such a manner as to place the LCAC in the center of a 5-by-6 matrix, where the sixth row/column is the surf. In this instance the surf acts as an additional row or column and moves laterally up and down the beach in association with the movement of



(a)



(b)



(c)

Figure 5 - Sea State Jumping The open ocean repeatable sea models are placed in a 5-by-5 matrix about the trainee such that the outer boundaries cannot be readily distinguished (5a). As the LCAC exits the center square (5b), the columns and/or rows furthest from the craft are effectively rearranged to form the 5-by-5 grid. This restructuring is eliminated in the direction of a shoreline (whenever surf exists) to preserve the sea-to-surf boundary (5c). Here, lateral reconfiguration is permitted where the surf is sixth row/column.

the LCAC. This method of rearranging the sea states in all directions except that of the shoreline is termed "lateral sea-jumping" (Figure 5b). By restructuring the tessellation grid in this manner, the trainee perceives a contiguous boundary between the sea state and the surf as the LCAC traverses to land.

The simulation of the surf areas for the Landing Craft, Air Cushion Full Mission Trainer proved to be a major hurdle in creating a complete oceanic environment. The problems of meeting the visual cues necessary for a generic surf zone were satisfied by introducing a massaged surf profile from the digitized data. The final surf scene contained the chaotic, violent nature expected by the hovercraft operators.

Support Ship Dynamic Wake

During an amphibious assault mission, the Landing Craft, Air Cushion is constantly bringing troops and equipment ashore. To minimize the time between successive beach deliveries, the LCAC operator must be able to efficiently enter/exit the well deck of a support ship.* The LCAC crew must work as a team during this maneuver to monitor the distance between the hovercraft to the support ship and adjust the encounter speed of the LCAC to the stern of the support ship.[2] Rough seas, fog, ocean spray, and nighttime well deck operations often make this a difficult maneuver. Instances of damage to the LCAC as it collides with the support ship during high seas are not uncommon. The visual cues modeled in the LCAC FMT include an illuminated well deck for day, night, or wartime operations, an animated signalman and moire lens to provide the LCAC with directional signals, and a flag mounted above the well deck to provide wind directional cues. Although the Navy agreed that these support ship attributes were beneficial during well deck entry missions, they did not provide the distance cues necessary for such operations. Only when a dynamic support ship wake was added to the visual scene was the crew able to consistently judge this distance.

Experience shows that a propeller-driven steaming vessel generally leaves a series of waves that are emitted from its bow, and an area of immense water turbulence at its stern. Wakes are a complex phenomenon that are generally affected by the sea state, wind, and ship's attitude.[7] In the LCAC FMT, a 10-knot steaming support ship dynamic wake was modeled. The bow wake, a V-shaped series of parallel waves generated as the bow of the ship slices through the sea, was modeled as a churning white animation directly at the bow of the support ship. Because the LCAC is seldom within view of the support ship bow (well deck operating procedures state that the LCAC must approach the support ship from the support ship stern), the bow wake proved to be a less significant cue and the details of the series of parallel waves from the bow were omitted. The churning froth near the bow was modeled to aid in identifying the ship as steaming when viewed from a profile perspective.

It was determined that the aft wake was a significant factor in judging the distance of the LCAC to the support ship, thereby creating a successful well deck scene. The aft wake was modeled as an area of turbulent, disturbed water directly behind the support ship created by the ships propellers. This wake contained a "collapsing" area of water to fill the volume of water displaced by the steaming ship's hull immediately at the support ship stern. The turbulence was modeled as a series of transparent animating polygons moving in a chaotic sequence at the ramp of the support ship (Figure 6). Unlike the sea state and surf, the stern wake was modeled as a chaotic froth of water. To maintain this froth-like appearance, the dynamic wake polygons did not change in contrast as the polygon normal became horizontal. As the wake moved away from the support ship, the turbulence became less chaotic and transparent polygons were used to model the wake. This allowed the wake to blend in with the sea.

The dynamic wake was built as a part of the

* - Each of the support ships contain a large platform, or "well deck", within its hull to load or unload various resources to or from the LCAC. The well deck is accessed from the support ship stern.

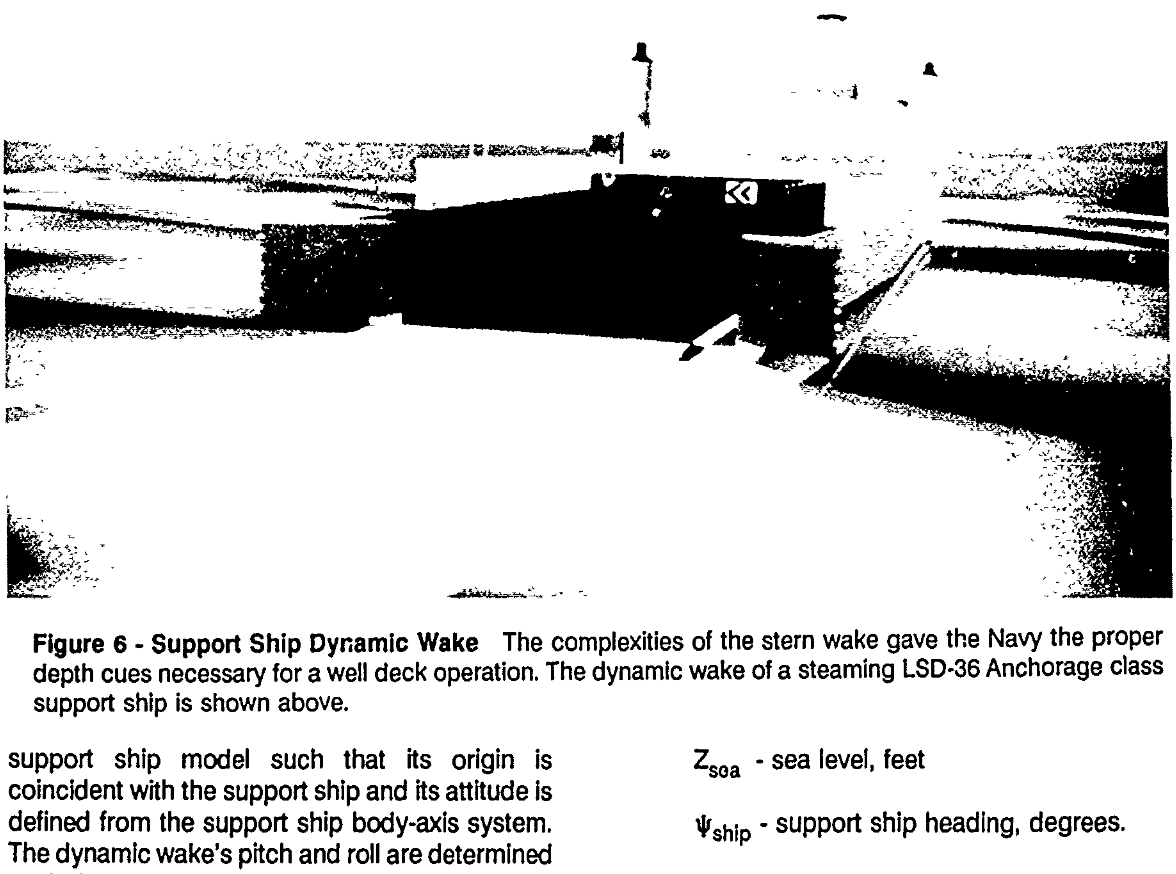


Figure 6 - Support Ship Dynamic Wake The complexities of the stern wake gave the Navy the proper depth cues necessary for a well deck operation. The dynamic wake of a steaming LSD-36 Anchorage class support ship is shown above.

support ship model such that its origin is coincident with the support ship and its attitude is defined from the support ship body-axis system. The dynamic wake's pitch and roll are determined such that the wake remains flat on the surface of the sea. The state vector of the dynamic wake is defined as follows:

$$w = \begin{bmatrix} X_{ship} \\ Y_{ship} \\ Z_{sea} \\ \Psi_{ship} \\ -\theta_{ship} \\ -\Phi_{ship} \end{bmatrix}$$

Where:

w - dynamic wake state vector.

X_{ship} - support ship longitude, feet.

Y_{ship} - support ship latitude, feet.

Z_{sea} - sea level, feet

Ψ_{ship} - support ship heading, degrees.

θ_{ship} - support ship pitch, degrees.

Φ_{ship} - support ship roll, degrees.

To observe an LCAC crew performing a well deck operation in high sea states is perhaps the most striking training aspect of the LCAC FMT. All of the hovercraft crew members must accurately perform their specific tasks, and the group must act as a unit to successfully complete this operation. The Craftmaster requires a correct visual scene as the Engineer and Navigator provide ownship state and distance information. The dynamic wake greatly increases the fidelity of the well deck operation and gives the LCAC crew the ability to perceive the proper depth cues required to perform a well deck entry.

Conclusion

Amphibious Task Forces are a cornerstone of the U.S. Marine philosophy of rapidly responding to global conflicts. In any analysis of Naval power, one fact should not be overlooked: virtually no coastal region on earth is immune from an attack from the sea. Weapon systems, flotillas, and specialized assault crafts would prove ineffective without proper training and familiarization with the capabilities of such systems. Without effective training, putting men and equipment ashore would prove an unpredictable and complex job. The Landing Craft, Air Cushion Full Mission Trainer allows an instructor to place a crew in a controlled environment where seaward hovercraft skills can be enhanced. The physical characteristics of the sea and surf are successfully modeled by a combination of mathematical analysis and on-the-spot creativity. By massaging the sea and surf elevation profiles to enhance the desired oceanic properties required by the Navy within the limits of the visual system used, Hughes Training, Incorporated has provided an oceanic environment suitable for hovercraft training.

References And Suggested Readings

1) Class, J., "The Implementation of the Landing Craft, Air Cushion (LCAC) Full Mission Trainer (FMT) Land Dynamics Reaction to Terrain", Proceedings of the 12th I/ITSC Conference, November 1990, pp. 144-149.

2) Donner, M., "The Simulation of the Dynamic Interaction of a Hovercraft Entering a Support Ship Well Deck", Proceedings of the 12th I/ITSC Conference, November 1990, pp. 136-143.

3) Fioravanzo, G., Naval Tactical Thought, Naval Institute Press, Annapolis, Maryland, 1979.

4) Ganguly, D., "Bangladeshis Hail Arriving 'Angels'; U.S. Marine Flotilla Joins Relief Effort for Survivors of Cyclone", Washington Post, May 17, 1991 section A, p. 20.

5) Graham, P., and Norton, J., "LCAC Medevac Considerations", Proceedings of Naval Review, Annapolis, Maryland, February 1990.

6) Grant, M. G., Oceanography, A View of The Earth, Prentice-Hall Incorporated, Englewood Cliffs, New Jersey, 1977.

7) Gross, M. E., "A Real Time Particle System for Display of Ship Wakes", IEEE Computer Graphics and Applications, May 1990, pp. 30-35.

8) Ladd, J. D., Amphibious Techniques, Learner Publications, Minneapolis, Minnesota, 1985.

9) Maroon, F., and Beach, E., Keepers of the Sea, United States Naval Institute, Annapolis, Maryland, 1983.

10) Moore, M., "Navy's 'Imminent Thunder' Assault Washes Out: Waves to Gunwales Deflate Air-Cushioned Crafts' Mock Attack on Beach Near Kuwait", Washington Post, November 18, 1990.

11) Schwarzkopf, H. N., Persian Gulf war press briefing by General H. Norman Schwarzkopf in Riyadh, Saudi Arabia. A complete transcript can be found in numerous sources, including: Washington Post, February 28, 1991, section A., pp 35-37.

12) Whitehouse, A. ,Amphibious Operations, Doubleday and Company, Garden City, New York, 1963.

Acknowledgments

The author would like to acknowledge the non-real time sea and surf elevation profile work done by ORI, Incorporated during the early stages of the LCAC project, and the work performed by Hughes Training, Incorporated, Arlington, Texas in the construction of the sea and surf polygon models. Their efforts were critical in the development of the LCAC FMT visual scene.

About The Author

Mark E. Donner is a Systems Analyst on the LCAC-FMT program at Hughes Training, Incorporated. He is responsible for the design, test, and integration of all visual related software. He has also contributed to the development of the dynamic software, particularly in areas related to the visual system interface. Mark has 5 years experience in his current position and holds a Bachelor of Science in Chemical Engineering and a Masters of Computer Science, both from New Jersey Institute of Technology.

ADVANTAGES OF AN OBJECT-ORIENTED DESIGN APPROACH TO THE SIMULATION OF LEADSHIP EFFECTS

Jerome M. Weiss, Section Head (Aero/Propulsion)
Ruth E. Korba, Systems Engineer
CAE-Link Corporation
Binghamton, NY

ABSTRACT

The B-2 Aircrew Training Device (ATD) employs object-oriented design (OOD) accompanied by the Ada programming language. The task of choosing objects to simulate vortex, bow wave, and engine exhaust effects of a leadship on the B-2 is presented from an OOD perspective. The B-2 software architecture of the leadship effects model created from an OOD approach is analyzed and compared to previously used software architecture at CAE-Link. These comparisons are made against architectures used in other military trainers. The trainers are evaluated in terms of maintainability and reusability. Conclusions are drawn as to which architectures are most efficient from a data concurrency/subjective evaluation and future applications perspective.

INTRODUCTION

The task of training a pilot to successfully navigate air disturbances caused by a preceding aircraft has been a goal of many high-fidelity simulators. The design of what is called a leadship or othership effects model has been approached differently throughout the years at CAE-Link, with the most recent model being that of the B-2 Aircrew Training Device (ATD). The B-2 ATD was the first to use an object-oriented design approach. Object-oriented design (OOD) is the organization of software into layers of objects, where the higher layers of objects usually relate to separately compilable software units. Its purpose is to emphasize maintainability and reusability of the trainer software.

This paper first introduces the B-2 ATD software architecture of the leadship effects model. The current B-2 leadship effects model, as well as its development, is explained from an OOD perspective. Next, the leadship effects software architectures for previous CAE-Link military trainers are discussed. These architectures are then evaluated in terms of maintainability, reusability, model fidelity, and complexity of software. This investigation also presents possible areas for improving the B-2 architecture. Such improvements could make the design more object-oriented and more efficient. The improvements are based not only on this study, but on lessons learned throughout the B-2 ATD design and test processes. Conclusions are presented concerning the effectiveness of these architectures as they relate to maintainability and transportability.

B-2 ATD OOD OTHERSHIP SUBSYSTEM

An othership subsystem was developed to satisfy the B-2 ATD training requirement for realistic aerial refueling and base escape characteristics. This subsystem supplies the vortex/downwash, bow wave, and engine exhaust characteristics of a vehicle preceding the B-2. The preceding vehicle (leadship or othership) could be a KC-135, KC-10, or another B-2. Such characteristics require the othership subsystem to determine the relative position between the leadship and the B-2 (lagship). This subsystem defines the leadship's and lagship's physical geometry. The leadship's effects are transmitted to the lagship by computing the delta forces and moments due to the leadship's presence. Different vortex/downwash models are used to compute these delta forces and moments depending on whether a base escape or aerial refueling maneuver is performed. The accomplishment of the B-2 ATD training requirements mandated that these models be of high fidelity. The details of this subsystem are presented in Ref. 1.

It was originally planned that the aircraft manufacturer would supply delta aerodynamic coefficient data describing the leadship effects on the B-2. Unfortunately, this data was not available from the aircraft manufacturer. Therefore, CAE-Link needed to take another approach. CAE-Link applied an OOD software representation in Ada code and current software engineering concepts, including generic modeling techniques. These concepts are discussed in the following paragraphs.

An OOD representation of a subsystem must satisfy two general requirements:

1. The subsystem should consist of self-contained objects that reflect logical/physical real-world entities. This supports the concepts of maintainability and reusability because all characteristics of the object are centralized. This allows ease of update, repair, and/or reuse by having these real-world entities modeled in one location.

2. The software implementation of each object will consist of a separately compilable unit or unit and its subunits. This allows model features to be selectable and therefore be more reusable. Troubleshooting design problems is made easier by having the objects coded in this manner. This also reduces the risk of accidentally affecting other software when these objects need to be modified.

The first step in our approach was to define the objects of the othership subsystem. Three object selection criteria were used:

1. The objects must be independent physical or logical entities. The things that describe the entities are called their attributes.

2. The objects must be such that changes to the B-2 ATD training requirements can be easily accommodated.

3. The level of decomposition of the objects must be governed by the real-world system operations, pure OOD theory, the data available to simulate the system, and the training requirements.

The othership subsystem was decomposed into three primary objects: leadship, air, and lagship. Each primary object satisfies the three OOD object definition criteria and translates into separately compilable Ada units. The Ada software configuration is depicted in Figure 1. This shows how each primary object was housed in separately compilable packages. The subsystem import and export interfaces with other subsystems were also made separate packages. The subsystem controller may contain computations but primarily contains logic code that calls Ada procedures within the various object definition packages. This subsystem controller determines if a leadship is close enough to a lagship to execute this code. The logic code then selects the correct procedures for aerial refueling or base escape training missions. Each object definition package contains Ada procedures relating to each primary object's attributes (see Figure 1). The details of the B-2 ATD software architecture can be found in Ref. 1.

Figure 2 is a block diagram of the B-2 ATD othership effects software architecture. This shows how the othership subsystem interfaces with the lagship forces and moments subsystem and gets leadship information from other subsystems.

Several software engineering principles were applied to each of the objects. These principles also support the goals of good OOD. The equations in-

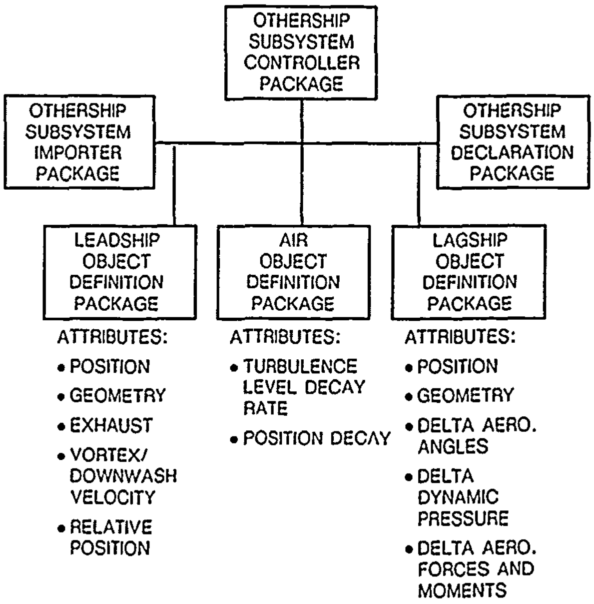


Figure 1 Othership Subsystem Ada Configuration

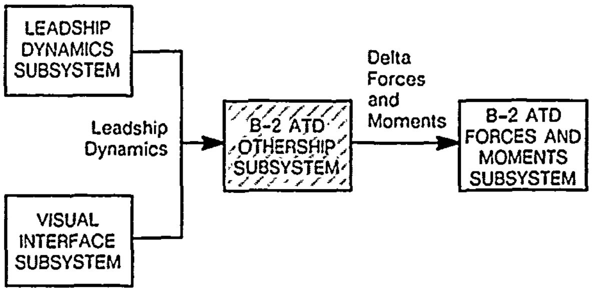


Figure 2 B-2 ATD Othership Effects Software Architecture

clude "adjustment constants" that allow instant investigation of problems with the pilot in the loop. The numerical definition of all application-dependent vehicle parameters is done in one location of the software. The definition of the vehicle parameters is accomplished in the declaration package. Equations were written and coded in generic rather than application-specific engineering notation. These principles allowed the coded equations to be independent of the vehicle being simulated, and increased their transportability and maintainability. The following two groups of equations are presented as examples of generic and application-specific engineering notations:

Generic Engineering Notation	Application-Specific Engineering Notation	
$L = qSC_L$	$L = 2400\ 0\ qC_L$,	where $S = 2400.0$
$Y = (\pi/4.0)b$	$Y = 74.6$,	where $b = 95.0$

With parameters b and S defined in one location, if an update becomes necessary, these parameters get modified only once. With the application-specific engineering notation, the parameters would be modified in each coded equation that uses them. The generic engineering notation also improves transportability because the equations and constants are more easily understandable by fellow engineers. These principles were applied to increase the software maintainability and reusability.

The specific use of the object definition criteria as they apply to this subsystem is discussed in the following paragraphs.

<u>Leadship</u>

This object models the preceding vehicle's geometry, vortex/downwash, and engine exhaust. All of this model's leadship information parameters and equations are housed within this object. The generic engineering notation was used within this object. This approach has the following advantages:

- The addition of a different leadship doesn't change the leadship equation. Their numerical values are defined in the declaration package.
- Problems identified during simulator testing or subjective evaluation related to the leadship can be addressed without risk of accidentally touching non-leadship software.
- Model features are represented by Ada procedures for each attribute. If a future trainer doesn't require a certain feature, that procedure is not included.

These advantages support maintainability and reusability. The attributes of the leadship object are listed below:

Leadship Velocity
Leadship Geometry
Leadship Position (Distance)
Leadship Engine Exhaust
Leadship-Generated Vortex/Downwash Velocity
Leadship/Lagship Relative Position

<u>Air</u>

This object models vortex/downwash decay factors due to the leadship/lagship relative position and atmospheric turbulence. All of this model's air information, parameters, and equations are housed within this object. The generic engineering notation was used within this object. This approach has the following advantages:

- Problems identified during simulator testing or subjective evaluation related to the air can be addressed without risk of accidentally touching non-air software.
- Model features are presented by Ada procedures for each attribute. If a future trainer doesn't require a certain feature, that procedure is not included.

These advantages support maintainability and reusability. The air object attributes are listed below:

Turbulence level decay rate
Position dependence decay relationship

<u>Lagship</u>

This object models the following vehicle's (lagship) reaction to the air disturbance due to a leadship. All of this model's lagship information, parameters, and equations are housed within this object. The generic engineering notation was used within this object. This approach has the following advantages:

- Transporting this model to different trainers would require no modification of the lagship equations – only the declaration package needs to be modified.
- Problems identified during simulator testing or subjective evaluation related to the lagship can be addressed without risk of accidentally touching non-lagship software. They can also be addressed directly because the various lagship attributes (features) are presented in different Ada procedures.
- Model attributes are presented by Ada procedures for each attribute. If future trainers don't require these attributes, the procedures are not included.

These advantages support maintainability and reusability. The attributes of the lagship object are listed below:

Lagship Geometry
Lagship Position (distance)
Lagship Delta Dynamic Pressure
Lagship Delta Aerodynamic Angles
Lagship Delta Aerodynamic Forces and Moments

The objects represent the key players in the real-world system of aerodynamic interference of a leadship on a lagship. The data available also fits the objects selected. For example, the leadship geometry data and vortex/downwash effects are given in terms of the leadship reference system (Reference 1) and therefore are housed in the leadship object. The leadship vortex/downwash and exhaust effects

are applied to the lagship by computing a delta angle of attack and dynamic pressures in the lagship reference system. Therefore these items, as well as the delta forces and moments, are computed in the lagship object. The relative position attribute is placed in the leadship object. This is done because the exhaust and vortex/downwash data needs this parameter in the leadship axis system. Placing relative position in the leadship object also reduces an interface between primary objects.

PREVIOUS LINK OTHERSHIP EFFECTS SOFTWARE ARCHITECTURES

The problem of simulating leadship (othership) effects on a lagship is not unique to the B-2 ATD. It has been dealt with on at least five different trainers. Each previous solution used a different functional approach coded in Fortran. The level of fidelity in each approach reflected that project's training requirements. A brief description of the five projects' (A, B, C, F, and S) approaches, subsystem software architectures in block diagram form, and general training requirements is presented below. The block diagrams can be compared to Figure 2, B-2 Software Architecture. The highlighted blocks in Figures 2 through 7 compute the simulated leadship effects. During the comparison, note that a software subsystem is similar to a module.

Project A

This project training requirement was to instruct the pilot on formation flying. This approach included simulating leadship vortex/downwash and engine exhaust effects on the lagship. This training requirement mandated that this model be a high-fidelity model. This approach was also coded in the application-specific engineering notation. It consisted of two primary modules which interface with the lagship's forces and moments module. The software architecture for Project A is given in Figure 3. The first primary module (Module 1) computed the leadship's dynamics, relative position between the leadship and lagship, geometry parameters, and leadship exhaust effects on the lagship. The second module (Module 2) computed the vortex/downwash effects and summed the vortex/downwash with the exhaust effects and passed the total delta forces and moments to the lagship forces and moments module. Module 2 used geometry parameters, exhaust effects, relative position, and leadship dynamics computed in Module 1. No data was supplied by the airframe manufacturer for this model.

Project B

Project B's training requirement was to instruct the pilot of the receiver in an aerial refueling maneuver. This approach included simulating leadship vortex/downwash and bow wave effects on the lagship. This training requirement mandated that this model be a high-fidelity model. This approach was also coded in the application-specific engineering notation. It consisted of two primary modules, plus pieces of code distributed throughout six other existing modules. The software architecture for Project B is given in Figure 4. The first module (Module 1) contained the leadship dynamics. The second module (Module 2) computed the relative position of the leadship and lagship. The six existing modules were the six aerodynamic coefficient modules. The leadship effects on the lagship were handled as components of the total six aerodynamic coefficients. Totaled aerodynamic coefficients were passed to the forces and moments module, as is normally done in Link simulations. Several outputs of Module 1 were passed to Module 2 in the form of direction cosines. Module 2's relative position was passed to each coefficient module to be used to compute the aerodynamic coefficient component due to the leadship vortex/downwash and bow wave effects. The airframe manufacturer supplied data in the form of aerodynamic coefficient components.

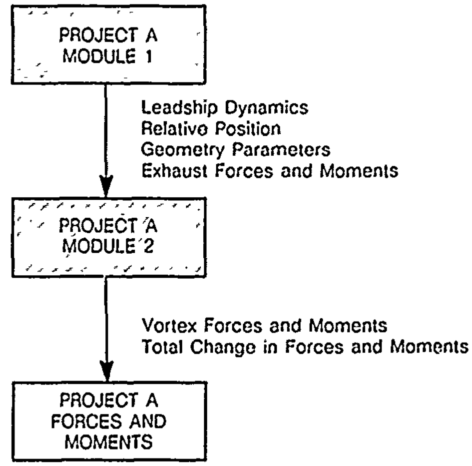


Figure 3 Project A Othership Effects Software Architecture

Project C

This project had two training requirements. The first was to instruct pilots of the tanker in an aerial refueling maneuver. The second involved having a leadship in front of the aircraft. Therefore, the pilot could be either a leadship or a lagship. The training requirements were such that the ownship would only need to experience a general effect when in the proximity of an othership. Therefore, Project C was a high-quality, low-fidelity model containing translational wind effects. This approach included simulating the vortex/downwash and bow wave effects of

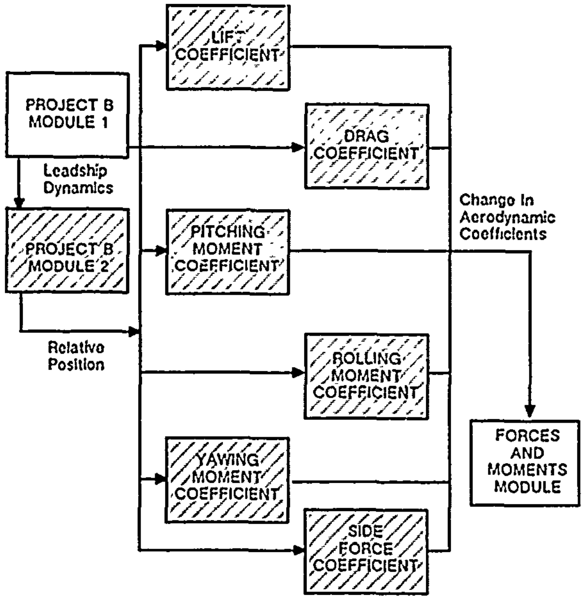


Figure 4 Project B Othership Effects Software Architecture

a leadship on a lagship. This model was coded in generic engineering notation with vehicle-specific data locally defined. The software architecture for Project C is given in Figure 5. It consisted of a single primary module which interfaced with the lagship equations of motion module. The primary module (Module 1) computed leadship dynamics, relative position, and delta translational wind components. These delta wind components were used to introduce the lagship effects into Project C's equations of motion. No data was supplied by the manufacturer for this model.

Figure 5 Project C Othership Effects Software Architecture

Project F

Project F's training requirement was to instruct the pilot of the receiver in an aerial refueling maneuver. The training requirements were such that the ownship would only need to experience a general effect when in the proximity of an othership. Therefore, this was a high-quality, low-fidelity model containing translational wind effects. This approach included simulating leadship vortex/downwash effects on the lagship. This approach was also coded in generic engineering notation with vehicle-specific notation locally defined. It consisted of two primary modules. The software architecture for Project F is given in Figure 6. The first module (Module 1) contained the leadship dynamics. The second module (Module 2) computed the relative position of the leadship and lagship. Both Modules 1 and 2 interfaced with the equations of motion module which computed changes in translational wind components. No airframe manufacturer data was supplied for this model.

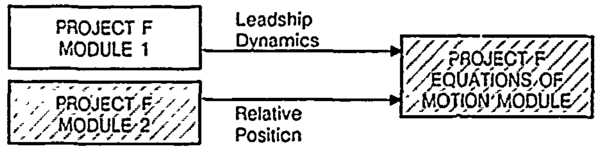


Figure 6 Project F Othership Effects Software Architecture

Project S

This project's training requirement was to instruct the pilot of the receiver in an aerial refueling maneuver. Although this project never implemented the leadship effects on the lagship, the planned approach is described in this paragraph. This approach would have simulated leadship vortex/downwash and bow wave effects on the lagship. It would have been coded in generic engineering notation. It would have consisted of seven primary modules, plus having code in one existing module. The software architecture for Project S is given in Figure 7. The first module (Module 1) would have contained the leadship dynamics. The existing navigation/communications module would have computed the relative position of the leadship and lagship. The other six primary modules (Modules 2 through 7) would have produced delta aerodynamic coefficients components due to the leadship and passed them to forces and moments to be incorporated with the other aerodynamic coefficients. No data was supplied by the airframe manufacturer.

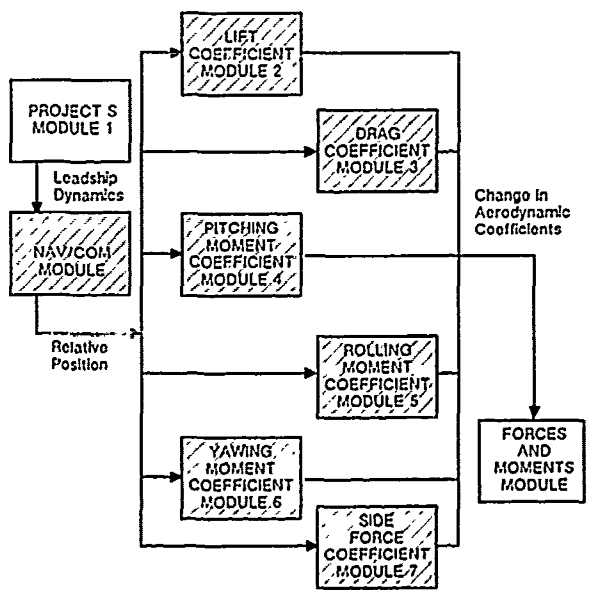


Figure 7 Project S Othership Effects Software Architecture

SUBJECTIVE EVALUATION

The software architecture of the B-2 ATD project was evaluated along with Projects A through S. Each project was considered a possible candidate for a new program with personnel unfamiliar with the software. Assumptions were that the languages and the high-level operating systems' architectures would be the same. The projects were rated on reusability or transportability, ease of update or maintainability, complexity of software, and model fidelity. A rating of good, fair, or poor was assigned for the first three categories, and high or low was used for model fidelity. Following is a description of the ratings categories.

The reusability category rates the ease with which software can be removed from one simulator and incorporated into another. Application-specific designs of code decrease an architecture's reusability. Examples of this include vehicle dependency and computational machine dependency.

The maintainability category rates the ability to change data or features. Data embedded in code is difficult to find and change. The addition of a new feature, such as a new leadship, or the removal of a feature, such as the base escape feature, is also difficult to incorporate if features are not organized separately.

The complexity of software category is a rating of how easy each project's code is to understand by a new user, and also the ability of any user to troubleshoot problems. Separation of features is an important part of reducing the complexity of software.

Model fidelity is not a rating but a training requirement. Some projects did not require a high-fidelity model. This category is included for insight into each project's design.

The subjective evaluation of the given software architectures is presented in Figure 8.

PROJECT	Reusability or Transportability	Ease of Update or Maintenance	Complexity of Software	Model Fidelity
A	C	B	B	HIGH
B	C	B	A	HIGH
C	B	B	B	LOW
F	B	B	B	LOW
S	B	A	A	N/A
B-2	A	A	A	HIGH

Key:

A - Good
B - Fair
C - Poor

Figure 8 Evaluation of Othership Software Architectures

LESSONS LEARNED

During the evaluation of the previous architectures and throughout the testing process of the B-2 ATD othership subsystem, several lessons were learned.

Two primary lessons were learned during the development and testing of the OOD presentation of the othership subsystem. The first was that all leadship attributes should be housed together. The B-2 ATD is a full Weapon System Trainer (WST) which has training requirements that necessitate the interaction of additional air vehicles. Early in the program, it was decided to include the leadship dynamics within the subsystem that modeled those additional vehicles. The leadship attributes should have been added to that subsystem instead of being placed in the othership subsystem. This has the following advantages:

- The leadship computations depend on the leadship parameters computed by the leadship dynamics housed in a different subsystem, therefore reducing interfaces.
- The lagship relative position to the leadship affects the leadship dynamics in the aerial refueling maneuver.
- Reduces duplicated definition of leadship specific parameters.

The second lesson was that the use of Ada procedures to correlate to subsystem attributes has the following advantages:

- Increases transportability because model features can be subtracted or added with minimal work. The training requirements of a simulator should govern what features are necessary.
- This allows extra flexibility if computational resources become a problem. Model features that don't have direct training value may be eliminated with minimal effort.

Additionally, there were lessons learned during the evaluation of the various leadship effects software architectures. The higher-rated architectures tended to have two general characteristics. Both could be considered good software engineering techniques:

1. The leadship effects were contained within compilation units outside of any other ownship models. Such centralization allowed a user to easily debug problems. This enhanced maintainability.
2. The leadship effects were modeled in generic engineering notation. This enhanced transportability and minimized errors to parameters used in multiple locations.

CONCLUSIONS

The conclusions reached by this study are based on the lessons learned and the subjective evaluation of various othership software architectures.

First, othership effects that are modeled and coded in separate compilation units yield high maintainability. This allows for quick problem isolation and faster, safer modifications.

Second, othership effects that are modeled and coded in generic engineering notation yield high transportability. This allows for quick changes of particular vehicles without affecting the inner workings of the models/code. Generic engineering notation also increases maintainability if data updates occur because they can be done quickly and clearly.

Third, training devices which do not require high-fidelity othership effects have less of a need for high maintainability and transportability. A design that is less object-oriented in such cases is admissible because of the lower likelihood of data updates and reuse. It would only be reusable on another project that also had lower model fidelity requirements.

Lastly, the OOD architecture was the only architecture that provided high reusability and maintainability for a high-fidelity model.

REFERENCES

1. Weiss, Jerome M., "Othership System, Software Detailed Design Document", Link SDDD BD100-0101-02, April 1990.

ABOUT THE AUTHORS

Jerome Weiss is currently leading the Flight Simulation effort on the B-2 Aircrew Training Device. His responsibilities include the aerodynamic, propulsion, and motion simulations. Previously Mr. Weiss led the aerodynamic simulation effort for the Link C-135B and B-52H trainers. Prior to joining CAE-Link in 1980, he was employed by Cessna Aircraft Company, Boeing Military Airplane Company, and the Air Force Flight Test Center. Mr. Weiss holds an MS in Mechanical Engineering from California State University at Fresno (1978) and a BS in Aerospace Engineering from the State University of New York at Buffalo (1973).

Ruth Korba joined CAE-Link Corporation in 1988. She has since worked as a Systems Engineer on the SR-71 and the B-2 simulator programs. On both programs, she was involved in aspects of propulsion and aerodynamic simulation. Miss Korba graduated from Syracuse University in 1988 with a BS in Aerospace Engineering.

SEMI-AUTOMATED FORCES: A BEHAVIORAL MODELING APPROACH

Hung T. Le, Ph.D.
Stephen E. Phinney
Vernon C. Seward

IBM Corporation, 9500 Godwin Drive, Manassas, Virginia 22110

Abstract -- This paper briefly reviews the problems and challenges faced by training communities in providing a realistic opponent for tactical training in a battlefield environment. A functional model for simulating semiautomated forces (SAFOR) is defined. Behavioral modeling using a motor schema approach is introduced for adversarial planning and navigation. A mathematical formulation using ellipsoid model is described. This is followed by a scenario developed for battlefield planning to reflect the applicability of motor schema instantiations for controlling semiautonomous agents.

1. INTRODUCTION

With today's increasingly complex tactical environment, innovative solutions to the associated problems of training will be required to prepare combat team readiness. This entails emulating opposing force and friendly force units to support combat training of battalion level armor units that include a network of manned vehicle simulators. In a battlefield training environment, these emulated forces must be deployed on a simulated battlefield in accordance with an established military doctrine. Training also requires effective control of emulated forces that incorporates the capability to direct emulated force tactics and movements across simulated terrain in order to conduct force combat engagements with other simulated units.

In light of the very complex nature of modern warfare, the requirements to effectively train military personnel, in many situations, stipulate a large number of entities to take part in a wargaming scenario. Using fully manned simulators, however, in a large force would be too costly in terms of equipment and human resources usage. In addition, it would be extremely difficult to find personnel who may be familiar with enemy doctrine and tactics to provide crews for manned enemy vehicle simulators. Thus, to alleviate the need for the full complement of a battle force and the role playing opposing forces (knowledgeable in enemy tactics) in a war-gaming environment, some research and development work has been initiated to build intelligent simulators that are capable of generating friendly as well as opposing forces, known as semi-automated forces (SAFOR), in a wide range of complex operational settings. While these research and development work is aimed at addressing problems in battle planning using artificial intelligence techniques, the environmental model has often been assumed static, and the sole application of expert system to automated battle planning is not sufficient to meet real time requirements for battlefield training environment. Problems in mission planning using optimal search techniques also do not lend themselves to real time applications. These shortfalls result in SAFOR systems that are highly constrained in their behavioral characteristics, and thus cannot adapt to emerging situations or changing environments as required in a realistic battlefield setting.

In this paper, we adapt a motor schema-based model, which describes the interaction between perception and action, to address problems in automated adversarial planning that provide active and reactive mechanisms for controlling semiautonomous agents. SAFOR's behavioral modeling concept with its functional elements will be presented in section 2. In section 2.1, a motor schema based approach will be introduced. This is followed by a mathematical formulation of ellipsoid models in section 2.2. Using this framework, we will discuss aspects of agent modeling, environmental modeling and group behavior in the context of adversarial planning in section 2.3. A number of features of the navigation code are given in section 3.

2. BEHAVIORAL MODELING

The goal of SAFOR modeling is to develop both the ability to simulate collective behavior of combatant agents to succeed in simulated battle engagements while requiring each agent to exhibit unique behavior and react to local terrain and battle conditions. Force simulation models currently tend to only model statistically collective behavior. This approach is weak in application such as some anti submarine warfare problems as well as in this application where the number of trainees and/or simulated combattants is undetermined. The approach used will result in an autonomous agent that can exhibit "local" behavior. The agent is linked to other autonomous agents via links emulating the actual human technology link between agents in a real battlefield situation.

A suitable simulation requires interchangeability between a human "operator" controlling the simulation agent and replacing the operator with an autonomous simulation. This approach allows the construction of engagement scenarios for the training of operator crews (i.e. tank tactics training) or evaluation of new equipment capabilities and tactics. To accomplish this, three aspects of the simulation are described below. the agent or combatant unit, the connection of these units into a coordinated group, and the governing environment in which the units operate.

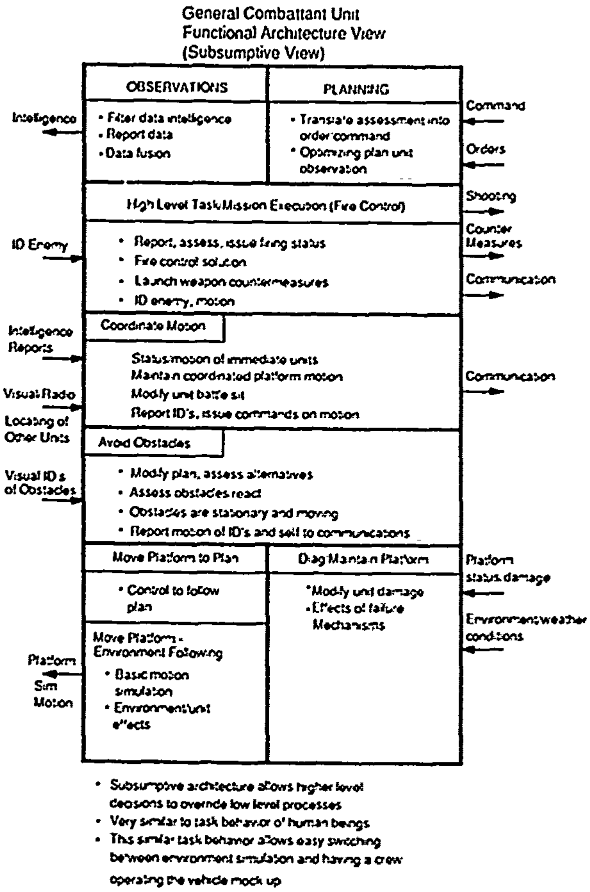


FIGURE 1. FUNCTIONAL MODEL OF A GENERAL COMBATANT UNIT

The primary objective, then, is not only to emulate the behavior of a real agent but to develop a general functional model whose instantiation can be used to simulate a variety of combatants from infantry and tanks, to submarines and sonobouys. In this way, this SAFOR approach can be the basis of a tank battle engagement. The major functional elements of this agent are shown in Figure 1. The functional elements, basic vehicle control (move the unit according to a plan and maintain the unit), obstacle avoidance, motion coordination with cooperating agents, military mission/engagement, and intelligence/planning are arranged in a hierarchical manner similar to a subsumptive approach (Brooks, 1988). The difficulty of using a pure subsumptive approach is that it only solves the reactive role of the agent whereas in our constrained simulation, our sensor world is ideal, behavior is sometimes dictated by a context driven tactic and overall objectives or plans to be met. This requires adapting the functional hierarchy to a task oriented approach similar to the one used by Shudy (1988). The agent simulation fulfills many of the tasks found in autonomous robot control systems (UNH for NEODTC, 1990).

The agent or combatant unit does not require a full robotic approach. The SAFOR environment is only a simulation with deterministic events in which few unexpected events can occur. Agent movements in the SAFOR environment also assume that terrain knowledge, gained from maps and multiple agent communications in the real battle environment, is common to the multiple agent simulation. This more easily constrains the planning function and simplifies path planning and execution.

The basis of the agent motion is a motor schema based model which models the unit's dynamical behavior based on the resolution of velocity potentials selectively applied to the agent at any one time. This requires the environment to be characterized by attractive/ repulsive potentials and unique feature designations to simulate the agent navigation around obstacles and react to snow, fog, visibility in day or night conditions, or follow a road. The higher level functions which are based on higher reasoning and plan execution will selectively override or modify the motor schema forces. These functions implemented as distributed processes will better apportion the processing loading.

To address interactions between multiple agents, the group behavior we are developing is based on the simulation of normal communications paths which exist in a battle force between cooperative units. Within the hierarchical military command, orders are disseminated "down" the command chain. Plan decomposition from the highest level to the lowest level produces collective behavior consistent with the overall expected results. Plans are based on externally imposed objectives, a defined environment, and perceived knowledge of the enemy configuration. Perceived knowledge is the result of fusing the individual unit perceptions up the chain of command to the highest level. Each simulated combatant, sensor unit, or commander can be represented by instantiating an agent. By moving intelligence "up" the communications link, passing plans "down" the link and using a common "Map"/ environment, the resultant group behavior will cooperate for a mission success. Likewise, the disruption of these knowledge sources will potentially cause a disruption in cooperative mission success. Hence realistic scenarios in the simulation are realized.

2.1 MOTOR SCHEMA APPROACH

The term "motor schema", originated in the fields of psychology and neurology, defines a generic specification of a computing agent. In other words, this concept of behavior represents an individual's response to its environment wherein each schema models a generic behavior. In this framework, the instantiations of these generic schemas provide the potential actions for the control of the computing agent.

The behaviors of an agent are represented by a set of schemas generated as artificial fields whereby actions and reactions are guided. Basically, an artificial potential is a mathematical description of the potential energy which may act upon an entity within a given environment. The potential may be associated with natural or man made entities at rest or in motion. Artificial potentials may be divided into two types: attractive and repulsive potentials. For each detected obstacle and/or region of threat, a modeled repulsive field is assigned. Likewise, a modeled attractive field is assigned for each target or objective on the battefield. These forces cause an entity to move through the environment in a manner directly responsive to the modeled velocity function of that environment. For autonomous navigation, attractive potentials may be assigned to objectives and repulsive potentials to obstacles to guide an entities navigation across a specified area. Thus, an entity may be directed to autonomously navigate toward a tactical or strategic objective while avoiding natural and man made obstacles of a moving and stationary nature.

In our problem, the motor schema model is used to determine the sequence of actions for achieving a set of military objectives. In this model, agents' behavior is be defined based on the velocity fields set up around obstacles and/or regions of high threats. To cope with the associated uncertainty with perception in the battlefield environments, a probabilistic mechanism is built into the velocity field generation. As the perceptual mechanism's confidence (activation level) exceeds a predefined threshold for action,

a repulsive (attractive) field is set up to surround the obstacle (target). The intensity of repulsive force is affected by the distance from the semiautonomous agents and the obstacle's perceptual certainty. Furthermore, the field strength may also depend on the type of mission, and therefore, can be set up accordingly.

The above described approach will provide the system with an ability to adjust plan dynamically during plan execution, as required in a dynamically operational environment with low predictability. Schema modeling also allows a large number of semiautonomous forces to be controlled simultaneously, and semiautonomous agents' behavior to be directed toward cooperative actions in a dynamic battlefield environment. Furthermore, with the use of schema models, some elements of stochastic behavior (random actions) are introduced, and therefore, agents' behavior will not be predictable. In addition, this approach also provides an effective way to model objects in the simulated terrain as well as potential fields, which gives rise to a very efficient algorithm.

2.2 SCHEMA GENERATION USING ELLIPSOID MODELS

2.2.1 ELLIPSOID MODELS

Any given object in the battlefield environment, whether stationary or moving, is represented by a set of ellipses (see for example, Yegenoglu et al. [1988]) denoted as a region of points enclosed by the contour $\| \Phi_0 \| = C$, where C is a positive scaling factor. In this formulation, the vector $\Phi_0(X)$ is defined as

$$\Phi_0(X) = -[\nabla H_0(X)]^{-1} H_0(X),$$

and

$$H_0(X) = \begin{pmatrix} (\alpha_{11}x_1 + \alpha_{12}x_2 + \beta_1)^{\sigma} \\ (\alpha_{21}x_1 + \alpha_{22}x_2 + \beta_2) \end{pmatrix},$$

where, $\alpha_{11}x_1 + \alpha_{12}x_2 + \beta_1 = 0$ defines the minor axis and $\alpha_{21}x_1 + \alpha_{22}x_2 + \beta_2 = 0$ defines the major axis of the ellipse. The lengths of the minor and major axes of the ellipse are given as 2C and $2\sigma C$, respectively.

To establish velocity fields for any target or obstacle, we require that: (1) the field be smooth and differentiable for smooth trajectories and constrained accelerations; (2) the field should point away from or tangential to the obstacle near obstacle boundaries; (3) outside the vicinity of an obstacle, the field should point toward a target; (4) the field must not contain any local minima; (5) the field should approach zero at the target boundary; and (6) the field should be additive to handle multiple contacts. To account for requirement (4), an "escape" velocity must be added to the field vector resulting from the interactions among fields contributed by targets and obstacles. The vector field can be then be computed as a linear combination of the attraction and repulsion vector fields. The resultant vector, which gives the directional force acted on the vehicle, is given as follows:

$$V(X) = \frac{\Psi(X)}{|\Psi(X)|}\left\{v\left(1 - e^{-b_a D_a(x)}\right)\right\}$$

where,

$$\Psi(X) = \Psi_a(X) + \Psi_r(X)$$

$$\Psi_a(X) = \frac{\Phi_a(X)}{\gamma_a D_a(X) \| \Phi_a(X) \|}$$

$$\Psi_r(X) = -\sum_{i=1}^{s} \frac{\Phi_{r_i}(X)}{\gamma_{r_i} D_{r_i}(X) \| \Phi_{r_i}(X) \|} e^{-b_r D_{r_i}(X)}$$

v is the velocity of the vehicle, s is the number of obstacles in the field of view, b_a and b_r are spatial repulsion decay rates, D_a and D_{r_i} are the distances to the target "a" and obstacle r_i, respectively. γ_a gives the class entropy for the set of attractors, and γ_{r_i} gives the class entropy for the ith set of repellers (Yegenoglu et al. [1988]). This implementation provides a means to cope with the associated uncertainty with perception in the battlefield environments.

The escape velocity can be computed as:

$$V_e = K_e e^{-b_e|\Psi(X)|} \eta(X),$$

where $\eta(X)$ is the unit vector perpendicular to $\Psi_r(X)$. The resulting vector can then be calculated as:

$$V_T(X) = V(X) + V_e(X).$$

2.3 PLANNING

2.3.1 Agent Modeling

The functional model for an agent or combatant unit, as shown in figure 1, depicts five behavioral layers required for the simulation of an independent agent. They are layered from the most fundamental tasks up to the most complex. Although this is the same notion as Brooks (1986) subsumptive architecture, this approach as a more constrained reactive schema (i.e. motor schema) vice reactive responses to blindly perceived obstacles. Imbedded in this model is the planning of a tactical response required by a military combatant.

The lowest level controls the agent in a constant motion. The motion is modified according to locally imbedded features (i.e. mud, fog, slopes). Overriding this behavior may be the constraint to follow a road or path. This constraint allows following a road even if it abuts an obstacle which could have a high repulsive potential. Also at this level, the agent readiness subfunction determines degradation due to equipment failure or firepower damage. Remaining energy resources, for examples fuel, battery, and so on, are evaluated and used by the planning function as a mission "cost".

The second level uses the motor schema model to selectively choose obstacles and modify the motion to avoid these obstacles. In addition, because these course changes deviate from the original plan, a secondary plan is invoked to negotiate the agent back to the original plan, if possible. The motor schema model is also used to perform this task. The identification of the motion of other "observed" agents and the motion of this agent is communicated to the next command level. This data, modified for simulated uncertainties, form the basis of data to be fused with other data at higher command levels.

The third level handles cooperative or group motion. This function executes group plans and controls the agent motion based on positions dictated by the group commander. The command levels maintain status on the cooperative agents and adapts their plans based on the changing motion of the individual agents in the group.

The fourth level encompasses the actual military mission outside the agents simulated ability to move about the terrain. This function contains the battle/engagement strategy and the control to simulate weapon firing. Even though these tasks are functional at a high level, the need for reactive behavior during battle requires a similar implementation as the reactive obstacle avoidance under the motor schema approach.

The fifth level contains the "links" to the rest of the cooperative agents and the support processes. The planning function translates assessments or intelligence into orders at a high command level or translates low level orders into

detailed action plans. The optimization of the plan for adapting to the agent's local environment is undertaken. The assessment or intelligence requires fusing agent contact information received from various agents and adapting the result to top level plan generation.

The functions and subfunctions included in this agent model can be instantiated, in some part, to most agents encountered in coordinated military operations. For example, the entire structure is used to simulate a tank. A mortar or sonobuoy requires some of the functions. The mortar's functionality is contained in the mission function and the sonobuoy's functionality is contained in the "Observations" or "Intelligence Gathering".

2.3.2 Environmental Modeling

The environment is set up as a generating function. The parameters imbedded in the representation of the terrain or ocean area are used to constrain the behavior of simulated agents. These parameters also serve as the basis for generating visual displays which are used in a trainer simulation where operators can replace an agent's simulation. The area is tesselated or segmented into regular geometric shapes such as squares. Each segmented area has characteristics assigned to it. The attractive or repulsive potential use in the motor schema modeling is associated with an area. This concept allows local constraints to be 'available' for an agent when it enters the area of the neighborhood of an area. Examples of these parameters are the local slope an agent must climb over or a small river bed the agent crosses over. Likewise day/night, fog, mud, snow, and rain are imbedded parameters that can be picked up by the agent when he enters the area and used to modify his behavior. An additional imbedded set of parameters relate to the 'cost' of travel across this segment or the threat weapon coverage . The cost parameters can be used to discriminate areas which are better or worse to travel over by modifying potentials in the motor schema model or aid in the decomposition of the strategic plan into detailed actions. Some of these parameters can change by themselves (i.e. day to night). In a highly parallel architecture, implementing the cells as cellular automata suggests efficient ways to manage the changing environment.

The terrain irregularities that the agent can drive over are part of the tesselated terrain whose irregularities an agent can't cross over but must go around become a terrain obstacle. These obstacles can be hills, rocks, or even rivers. The obstacles will be simulated as forms of elliptical repulsive velocity potentials.

2.3.3 Group Behavior

2.3.3.1 Cooperative Motion

The next level of challenge beyond a single agent successfully navigating the terrain is to simulate a cooperative motion among agents. The goal in agent cooperation is to coordinate the motion of many agents to result in the successful execution of a group plan. This coordination is dictated by communications from commanders down the command hierarchy. A typical arrangement of a group of cooperative agents is moving within an ellipse. By setting up a potential about a lead agent which is a neutral potential with a boundary potential wall, the agents within the elliptical wall will move in the direction and speed commanded by the lead agent as well as navigate locally avoiding collision with other cooperating agents. This elliptical wall defines the "influence" from an agent. It is sufficient for higher levels of command to know that his agents are with an elliptical boundary and not necessarily where each are specifically located. A statistical assessment is sufficient. This treatment of cooperative motion is consistent with the planning each level of command and how course or fine his knowledge of the distribution of his commanded agents is.

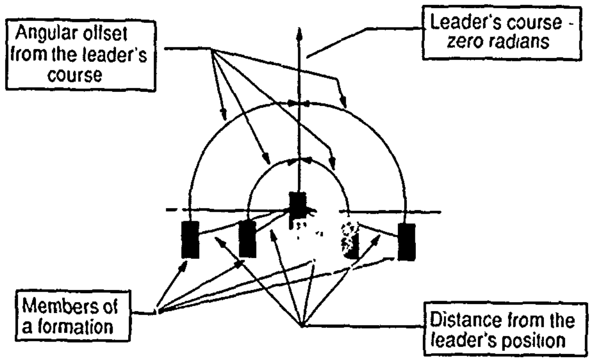


FIGURE 2. COOPERATIVE MOTION

Within the ellipse, each member agent is commanded to maintain a position consistent with a position vector assigned to it by the lead or commanding agent (figure 2). Agents or vehicles following a formation also follow the configuration dictated by the appropriate position vector. Agents avoid collision with other members of the group because each is an independent agent with his influence region creating reactive plan modifications.

Another approach to implement a march in formation is depicted in Figure 3. A formation is represented as a template in which the position of each vehicle is defined as an attractor. In this configuration, when vehicles have to break formation to avoid a moving (or stationary) obstacle, the attractors will force the vehicles back to their original locations. The force, which acted on any one vehicle, is a function of the distance from its original position. The farther the vehicle is away from its formation template, the faster it will move toward it until it gets back in formation, where the force (acceleration) acted on the vehicle will decrease to zero.

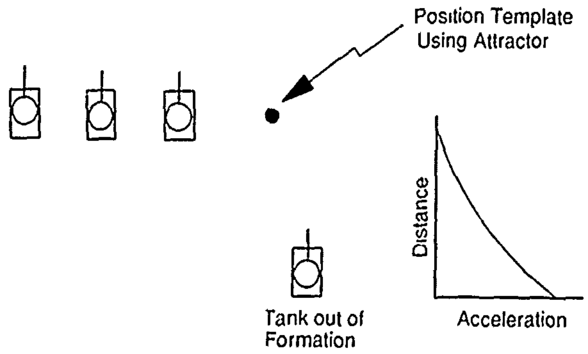


FIGURE 3. REGROUP IN FORMATION

Figure 4 shows an example scenario of cooperative agents moving along a road imbedded in the environment. It is expected that because each agent determines its minute-to-minute reactive behavior, the influence potential drop off near the agent will produce traffic jams and grid-lock as we see on today's major metropolitan highways as more and more agents are crowded on the road. Other agents not a part of the cooperative group will be treated as moving obstacle, friend or foe, which the member agents must individually decide whether to outrun the obstacle or wait for its passing before resuming the original cooperative motion plan.

- Vehicle A has mountain, road, and vehicles B and C in field of view; plan says: follow road --> ignore mountain, avoid vehicles B and C.
- Vehicle B has road in filed of view; plan says: Follow road --> carry on as planned.

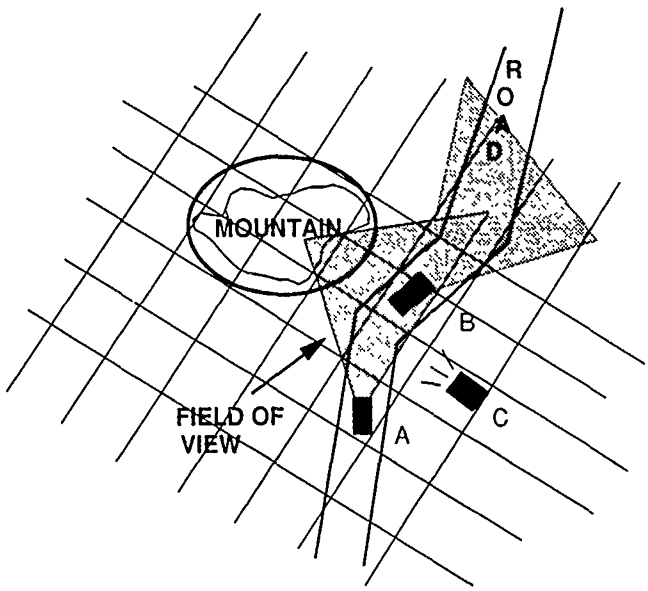


FIGURE 4. EXAMPLE SCENARIO

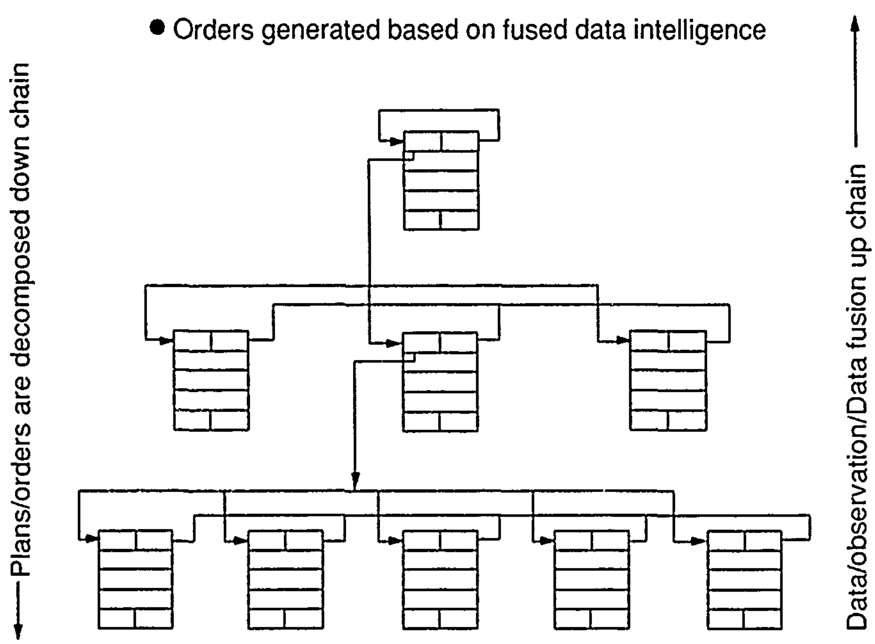


FIGURE 5. COMMAND/COMMUNICATION FOR GROUP LINKING OF COMBATANT UNITS

2.3.3.2 Hierarchical linking

The total command structure where each cooperating agent is linked by simulated communications to the others is shown in figure 5. In the figure, each agent is represented by a symbolic function diagram (figure 1). Messages are passed up the communications link which is the intelligence from each low level agent being fused and interpreted as it moves up the chain. At the top level, the simulated command officers take the intelligence, evaluate it against the plan, and generate new orders. These orders are disseminated down the command chain. At each lower level, the orders are further decomposed into detailed plans for the lowest level agent. To reinforce cooperative motion, each simulated agent has a terrain knowledge because in the real world, each combatant has a map. The particular simulation of these communication paths correspond to radio nets, or even sonar links in an underwater environment.

3. Implementation of Navigation Algorithm

To demonstrate the overall concept, the Moving Platform System (MPS) developed at the Naval Postgraduate School (Zyda et al., 1989) is used as a simulator testbed to conduct performance study of the motor schema-based approach as applied to navigation. The MPS provides a good vehicle to experiment with the navigation algorithm in that it allows for the creation and control of several types of vehicles on a simulated terrain.

In this implementation, some of the features that are added to the MPS as a mean to facilitate man-machine interface controls for demonstrating the navigation code are:

- The ability to select the repulsive field strength for each vehicle. Field strength is measured by how close an opposing vehicle will get before its course is affected. Four repulsive field strengths are currently provided, 1 kilometer, 1/2 kilometer, 1/4 kilometer and 0 kilometer or no field.

- Selecting whether a vehicle will 'sense' another vehicle's repulsive field. This feature will allow a vehicle to have a field but be oblivious to other fields.

- Providing multiple targets for each vehicle. This feature allows each vehicle to have a list of targets independent of each other. Multiple targets allow each vehicle to have a 'mission' that can include moving to several positions. Once the final target is reached the vehicle stops.

- A formation can be created with any combination of vehicles and in any position. Any vehicle within a group can be selected as a formation leader. If this option is selected, each subsequent vehicle generated will be a member of the formation.

With the navigation algorithm in place, autonomous control of numerous vehicles can be realized. Obstacle avoidance is achieved by establishing repulsive field around objects (threat regions, moving or static obstacles) in the simulation. The autonomous vehicles sense the strength of the fields and turn away when the field strength reaches a predefined level. An attractive field can also be generated for specifying a target toward which the vehicle is steered. The combination of attractive and repulsive field produce a vector on which the vehicle will steer while moving through the simulated environment.

4. SUMMARY

In this paper, we presented a motor schema based approach to modeling SAFOR's behavior. Using a subsumptive architecture, we developed a hierarchical structure for SAFOR that exhibits major functional elements of a combattant unit. These functional elements include basic vehicle control, obstacle avoidance, motion coordination with cooperating agents, military mission/engagement, and intelligence and planning. Within the very levels in SAFOR's hierarchical structure, we illustrated the applicability of the motor schema approach in basic navigation, cooperative or group motion in battle march using ellipsoid model, and low-level planning in a battle engagement scenario.

5. REFERENCES

Brooks, R. (1986), "A robot layered control system for a mobile robot," IEEE Journal of Robots and Automation, RA-1.(1), pp. 1-10.

Semet, H., Fujimura, K., (1988), *Path planning among moving obstacles using spatial indexing*, Chapter 1, pp.1662-7.

Shudy, R., Duarte, C. (1990) Advanced autonomous underwater vehicle software architecture.

Yegenoglu, F. and Stephanou, E. H. (1988), "Real-time obstacle avoidance in dynamic environments," to appear in IEEE Transactions on Automatic Control.

Zyda, J. M., Fichten, A. M., and Jennings, H. D. (1989), "Graphics workstations and 3D visual simulation: Some performance expectations and measurements," Technical Report NPS52-89-046, Monterey, California: Department of Computer Science, Naval Postgraduate School.

UNH for Naval Explosive Ordinance Disposal Technology Center (April 1990). Systems/Subsystems investigation for a multisensor.

Autonomous Underwater Vehicle Search System. Final Report NAVEODTECHCEN technical report TR-301.

6. BIOGRAPHIES

Dr. Hung T. Le has been involved in the research and development of image processing algorithms for underwater acoustics applications. More recently, he has been working in research and development of AI/Expert systems for training.

He has published numerous papers in the areas of statistical signal processing and artificial intelligence. He currently serves as an adjunct professor in the department of Operations Research and Statistics at George Mason University.

Dr. Le is a member of Tau Beta Pi, Eta Kappa Nu, the Institute of Electrical and Electronics Engineering, the Operations Research Society of America, and the American Statistical Association.

Stephen E. Phinney received a B.S. in Physics at the University of Rochester, New York in 1964. He is an Advisory Systems Engineer for IBM, Federal Sector Division, Manassas, Va. He has been developing signal processing and system architecture requirements for submarine sonar and combat systems for AN/BQQ-5, Trident, and AN/BSY-1 combat systems. He was responsible for Sonar Trainer design and environmental simulations. More recently, he has been responsible for site AI technology introduction. He is involved in research and development of AI/ Expert systems for training and control systems for AUVs and robotics. In addition, he is involved in the development of expert systems diagnostic applications in Naval systems development and operational environments.

Vernon C. Seward has been working as a programmer in linguistic and office systems applications with the Application Solutions Division at IBM Bethesda, Maryland, since 1989. He is currently conducting research and development of Artificial Intelligence systems in training and simulation. His responsibilities include design and development of AI algorithms for tactical training systems.

MODELING OF THE INTELLIGENT THREAT IN A DENSE TACTICAL TRAINING ENVIRONMENT

By Stephen J. Hunter
&
H. Reed Puckett
AAI Corporation
P.O. Box 126
Hunt Valley, Maryland, 21030-0126

ABSTRACT

The proper training of aircrews requires a wide variety of subsystems such as visual, radar, g-cuing, etc. A subsystem gaining more and more emphasis in today's trainers is the tactical training environment simulation. Within that subsystem has emerged the realistic modeling of the intelligent threat. The simulation required includes a representation of the threat including its platform, sensors, emitters, and its weapons plus the decision making process and intelligence of the threat operator. Tradeoffs must be made between the need for high fidelity simulation and the computer resources available. Some of the tradeoffs include threat simulation capacities and selection processes, weapon simulation fidelity, Electronic Warfare (EW) sensor simulation fidelity The degree of simulation of the threat's decision making process and tactics is also important and may be accomplished in several ways. In the A-6/F-14 Trainers, for example, a threat reaction algorithm process is used to simulate the actions and reactions of a threat to the total environment. Such algorithms are used for each threat element such as the threat platform, sensors and weapons to create a composite overall interactive threat behavior.

INTRODUCTION

The training of aircrews in modern simulators frequently requires the realistic representation of a dense tactical environment with the simulation of threat systems that intelligently interact with the rest of the tactical situation especially including the trainee's actions and reactions. Threats include weapon systems such as guns and missiles which might harm the trainee (or prevent him from performing his mission) as well as the sensors (such as radar) which support the weapon. The threat also includes the platform (aircraft, ship, vehicle or fixed site) which carries the weapon and/or supporting sensors, plus command and control and communications equipments.

A threat model needs to simulate the represented threat and its components (platforms, sensors, weapons and operator) as realistically as possible including their capabilities and limitations. Other threat capacities to be modeled include the capabilities and limitations of the threat platform (speed, altitude, turn rates, range and etc.) the threat sensor signal parameters for each mode of operation (acquisition, track, engage and etc.), transitions which occur in signal characteristics during threat operations, and the threat system's action and reaction times. Realistic modeling of weapon characteristics and limitations are also an important element of threat simulation.

In addition, the realistic representation of the thought process, decision making process and reaction times of the threat operator is important in providing the capability to teach the trainee tactics which would he would use in countering the real threat.

The software designer of the simulation of these threat elements must also strike a balance between fidelity and the consumption of computational resources. This paper provides an overview of these threat elements and their characteristics as simulated in a typical trainer. The description is based upon the AAI-developed tactical software used on the A-6/F-14 Aircrew Trainer Suite (representing the A-6E/SWIP and F-14D aircraft configurations).

THREAT ENVIRONMENT

An important feature of an Aircrew Trainer system such as that for an A-6 or F-14 aircraft is the simulation of the tactical environment. Modern aircraft training systems include the capability to represent enemy threats including surface (ground and ships) based anti-aircraft weapon systems and airborne air-to-air weapon systems.

The simulation of the tactical environment must be dense enough to represent real world war time situations. The A-6/F-14 Trainer Suites can simulate up to 120 active threats at one time. These threats are represented by simulated fixed or moving platforms each with one or several radar or other sensor systems. Some of these "threats" may not have weapons but rather represent threat support systems such as early warning radars, target acquisition radars, command and control systems, standoff Electronic Countermeasures (ECM) systems, and etc. However, most of the "threat" emitters are associated with

gun and/or missile weapon systems.

Often the simulation of a threats responses may include the capability to represent various degrees of training, readiness and proficiency of the threat operators. The A-6/F-14 Trainers simulated threats can also be programmed to have one of four different skill levels (from beginners to "Top Gun" level). It also might be noted that the threats, their platforms and weapons plus their dispensed chaff, flares and decoys can constitute as many as 400 or 500 objects being modeled at any one time.

THREAT SELECTION

Selection (Figure 1) is a technique where certain threats are selected for processing by various portions of the simulation models. This permits the available computational resources to be applied to that portion of the threat environment which provides training benefits at any given point in time of a training mission. Threat objects which are too far away to be detected by the ownship Electronic Warfare (EW) sensors or are blocked by terrain features, need not be further processed by sensor simulation software, for example.

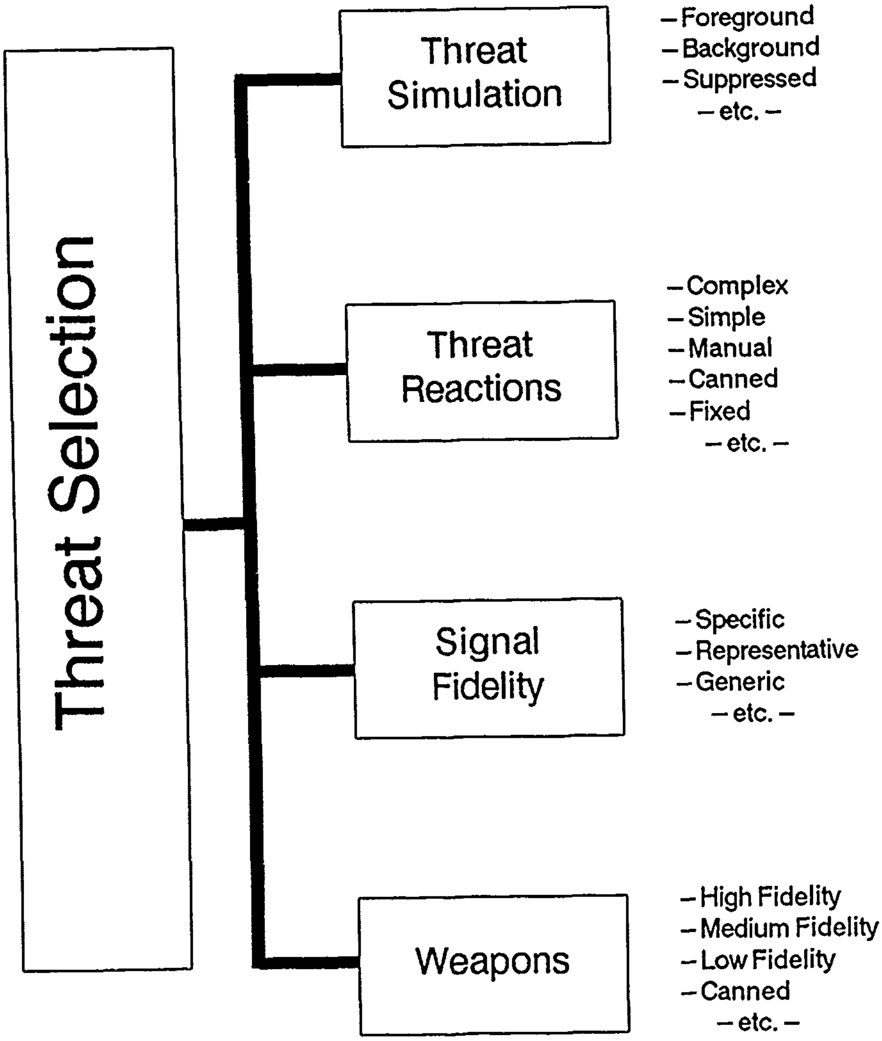


Figure 1 **Threat Selection Process**

Various degrees of simulation fidelity may also be appropriate for those threat objects which are selected for simulation in the environment (Figure 2). A minimum of fidelity, dynamic interaction, and the like, is required for threat objects remote from the trainee's ownship which interact with and affect the ownship the least (and form a "background" in the environment for the trainee). Other threats nearest the ownship and interacting more intimately with the trainee (perhaps attacking the trainee or being attacked by him) require the highest fidelity and degree of simulation. In the A-6/F-14 Trainers, for example, the 12 closest threats most involved with the ownship are simulated as fully "interactive", while more remote threats are to some greater degree preprogrammed in terms of their interactions with the tactical environment.

Another example is the selection of weapon model fidelity. In the A-6/F-14 Trainers the weapons fired by threats and the ownship may be modeled with either a high fidelity or a low fidelity weapon flyout model. The flyout calculations are used to determine weapon success or failure and resulting damage to the target. A few high fidelity models are run, at any one point in time, only for those weapons directed at or fired by the trainee's ownship. Lower fidelity models are used for modeling weapons launched by others against yet other "third parties". These lower fidelity models use simplified algorithms and reduced update rates to conserve processing resources.

OWNSHIP SENSOR SIMULATION

For aircraft like the A-6 or the F-14, Radar Warning Receiver (RWR) Electronic Warfare (EW) signal sensor is the main device used for detecting the presence of and identifying the type of threat systems in the tactical environment. The threat system includes sensors such as radar systems for detecting targets and aiming and guiding weapons. These radars are called threat emitters since they emit radar electromagnetic signals. A threat platform (such as an aircraft, surface ship, vehicle or ground site) might be associated with one or several radar emitters, and these, in turn, might each have one or several signals.

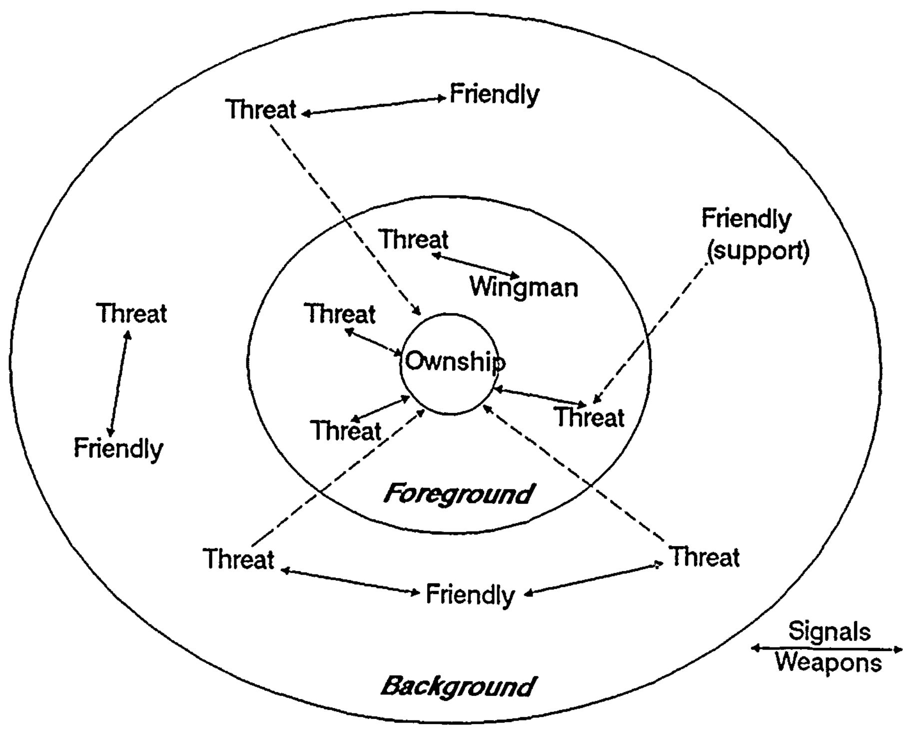


Figure 2 **The Threat Environment**

These signals each have unique signal characteristics which can be described in terms of signal "parameters". Parameters are values of signal characteristics such as signal power, signal frequency, pulse repetition frequency (PRI), pulse widths, antenna gains and antenna motions (or "scans"), and etc. Often the emitters have different "modes" like "search" and "track" modes. The signal characteristics (such as antenna scans and/or PRIs) will normally have some difference in parametric values from mode to mode which, in turn, might be detected to allow the emitter mode to be displayed by the RWR. An RWR might also provide audio outputs detected from the threat emitter signal pulse trains. These can provide a further warning to the trainee and might also be used to further identify the emitter type and activity.

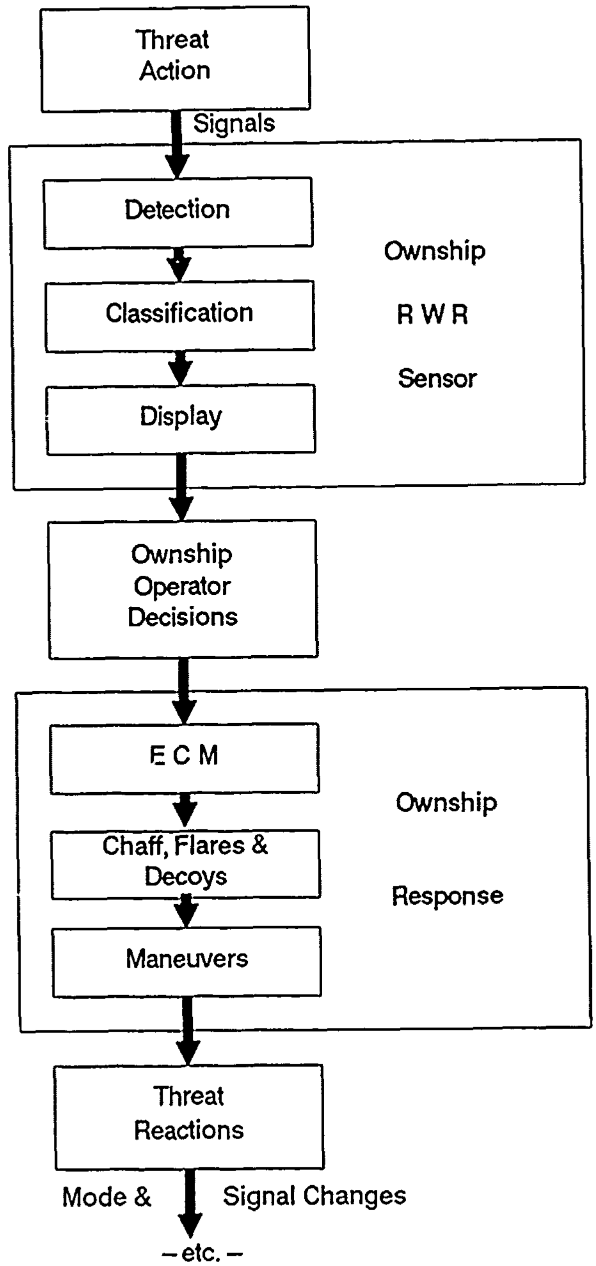
The F-14D aircraft contains an AN/ALR-67 Radar Warning Receiver, for example, which detects and classifies these signals and displays the type and relative position of the threat to the aircrew. The simulation of the RWR in the trainer involves modeling the real RWR signal detection and processing vs. the received threat signals. The simulated threat environment must therefore provide the signal parametric data for the active threats in their appropriate mode to the trainer RWR models in order to achieve a realistic RWR threat response simulation.

The signal parametric data is also used by the simulation models for the ownship ECM equipment which respond to detected threat signals (plus the trainee control inputs) and radiate countering "jamming" signals. These countermeasures might prevent a threat from being able to maintain a track condition on the ownship and, if so, would cause a "mode" change of the threat signals from, perhaps, a track mode to a search mode. These threat reactions need to be included in the threat simulation to realistically represent a threat environment.

Note the nature of a sequence of threat and trainee action, reaction, further reaction, and etc., which is quite representative of the threat environment and ownship equipment (and trainee) interplay (Figure 3). These steps in the sequence of actions and reactions are also separated by response times from a fraction of a second to a few seconds or even minutes. Variations occur due to equipment limitations, operator preferences, skill levels, errors, oversights, and etc., as well. A realistic threat simulation should include such variations in responses from one type of threat to another as well as for different examples of the same threat.

THREAT REACTIONS

In order to model changes in a threat's activity in response to its environment two things are needed: data and logic. The data provides information about the capabilities and status of the threat and about items in the environment. Examples might be the frequency of an signal, the mode of an emitter's operation, direction of travel, speed, and altitude of a moving object, and etc. "Logic" is needed to decide what to do about the situation represented by this data. Various approaches have been used for this logic including manual control by instructors or operators of various items in the environment. For a manual control approach, the available relevant data is presented to the operator so that he can decide what the item he is controlling should do. He enters control inputs to cause the simulation of the object to proceed in response to his decision (such as to change heading, for example).



Actions & Reactions

Figure 3

Control of threat objects using AI (Artificial Intelligence) approaches is also popular. Basically, the decision making process of the real threat operators are represented by AI models. AAI utilizes a threat logical control approach (for the A-6/F-14 Trainers, for example) which are called "reaction algorithm" models. These algorithms are prepared in an reaction algorithm language which uses terms familiar to the trainer instructors. This allows them to prepare or update algorithms without having special knowledge or training in computer programming languages and techniques. Basically the statements in the algorithm contain tests for checking some part of the data describing the environment followed by directions for actions (or reactions) to occur when some condition defined by the test is satisfied. The direction for an action can also be defined to include a reaction time or a range of reaction times.

Usually an overall threat algorithm is really a collection of several algorithm parts, one for each threat mode. Only the portion appropriate for the present threat mode is processed during any one algorithm update. One of the actions which the algorithm can cause, is a change to another threat mode ... which in turn causes another part of the algorithm to be processed during later updates. These actions also effect the threat data bases, such as the data describing a threat's mode of operation, its signal parameters, and so forth. These data changes then, in turn, may cause reactions from other algorithms for other threats when they are subsequently processed.

An example might be for threat system which is presently in the search mode. A part of the search mode algorithm might define that if an in-bound target is detected within a certain azimuth sector assigned to the threat, that the threat will shift to a track mode, tracking the detected target object after a delay time of, perhaps, 3 to 15 seconds. When this algorithm is processed and it is found that a target matches the defined conditions, a time is randomly selected between 3 and 15 seconds. After that time (if the conditions still hold) the threat mode is changed to a track condition. This causes the threat signal characteristics to change to those values appropriate for the track mode, and subsequent updates of the threats algorithm will then use the statements for the track mode rather than the search mode. The track mode might then have further tests to determine if and when a weapon might be launched which might, in turn, result in a shift to a "launched" mode of the threat. That mode then would probably have logic to return to the search mode if the target is destroyed (or if it escapes).

The algorithm for a particular kind of threat system may also have statements which control the threat reactions as a function of the skill level of the threat being simulated. The skill level can be defined as part of the training mission data for a particular threat. It might also be changed by the instructor during a training session.

Typically (including for the A-6/F-14 Trainers) the capability is included for the instructor-operator to take control of a threat. When the operator selects a manual control condition, the threat remains in the state of operation (or "mode") which it last had under the algorithm's control. The algorithm would no longer be processed, however, and the threat would not change modes except when manually commanded to do so by the operator. Threat changes made manually also affect the environment data base, which continues to be reacted to by the algorithms of other threats. The operator can also return a threat to automatic threat reaction algorithm control, which then begins processing in the last manually selected threat mode.

The above is an example for basically the radar system of a threat system. Such a radar system might be located on a moving platform (such as an aircraft or ship). The complete threat might then consist of the platform, a radar system (or perhaps several radar systems), and (one or more) weapon systems. Separate reaction algorithms would be associated with each threat element, i.e., the platform, each radar system, and so forth (Figure 4). Each of these would have sub parts such as for each radar mode, each platform activity mode (like "orbit" and "attack") and the like. Each algorithm piece processed can result in changes which effect the data base representing the state of the platform, radar, weapon, and etc. These, in turn, might effect the actions of the other algorithms for the other elements of the threat, as well as for algorithms for other objects in the environment.

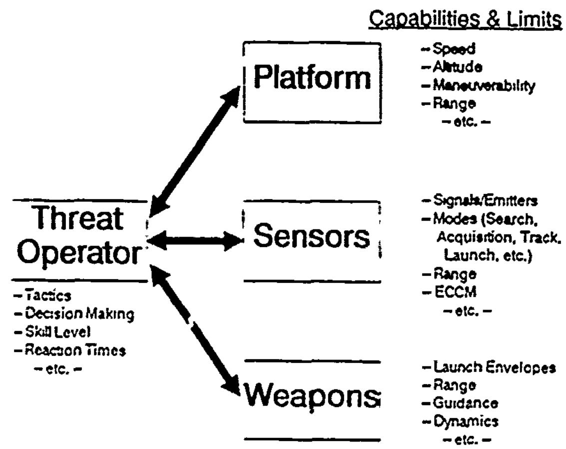


Figure 4 Threat Model Elements

Thus the algorithms contain logic which can interplay with each other as well as with instructor-operator manually controlled objects. Of course this environment also interplays and interacts with the simulated ownship and its equipment controlled by the trainee. Thus the purpose of the algorithms is achieved, namely to provide tactical training to the trainee by the simulation of an intelligently reacting threat environment.

Each type of threat has its own particular reaction algorithm. The same algorithm is processed several times each update if their are more than one example in the simulated environment of that type of threat. Separate data is maintained for each example of the threat, of course. It also is possible to have several different variations in a threat algorithm available to represent different behaviors or tactics from one threat example to another.

COUNTERMEASURES EFFECTIVENESS

An important part of the environment and tactics is the effect of threat countermeasures. These effects include the impacts of jamming signals which might interfere with radar sensors, the dispensing of chaff and flares which might confuse radar and IR sensors, and the results of threat avoidance maneuvers. Some of these effects create the appropriate result "naturally". For example, a threat weapon flyout model may not be able to track a target when in performs evasive maneuvers, and therefore the modeled missile will miss the target. Other effects might be accomplished by providing countermeasures data in the environment data base and utilizing threat reaction algorithms to cause appropriate threat reactions. In the A-6/F-14 Trainers, for example, jamming effectiveness models determine the impact of each jamming source (especially ownship jamming) on any threat it might affect.

The jamming effectiveness modeling involves solving the appropriate form of the radar range equations to determine threat radar resultant jamming to signal (J/S) ratios. A jamming type identification combined with tables of jamming type vs. threat type effectiveness values plus the J/S ratio are utilized to determine a magnitude of jamming effectiveness for each jamming source vs. each threat. This data is then available to the threat algorithms and might cause radar "break locks", for example. That is, it might cause a radar to lose track on a target and return to a search mode. Jamming might cause an increase in range and/or angle tracking errors, or might prevent the radar system from being able to range and/or angle track a target.

Effectiveness calculations are further complicated by the need to support not only self protection jamming but also standoff jamming as well. The form of the radar equations used vary for these two cases, and all combinations of threats and jammers must be considered. Jammer vs. threat radar antenna positions and orientations are particularly important in the J/S ratio determinations, and varies for each threat-jammer combination considered. The point is that the jammer effectiveness is not a universal piece of data which may be used by each and every threat algorithm. Effectiveness data must be developed for each threat-jammer combination. However, only those combinations which might interact (e.g., those using the same frequency range) need be considered.

NETWORKS

Combinations of threats may communicate and cooperate by sharing information and coordinating their activities. An example would be an air defense network consisting of a number of early warning search radars, height finders, and anti-aircraft missile and gun weapon systems. The information obtained from the early warning radar systems would be used to coordinate target assignments thereby avoiding duplications, etc. Further this data is used to ready the weapon systems for the target approach thereby reducing weapon system acquisition and reaction times and increasing weapon success probabilities.

An approach which has been used by AAI on the A-6/F-14 Trainers for simulating some of the effects of such a threat group cooperation is the use of "networks". The instructor may define (during mission preparation) a group threats which will share information as a network. Individual threat algorithms take into account the existence of target detection by any member of a network. Such information is assumed to be passed on to the individual threats of the network allowing a reduced target detection, identification and reaction time by a threat when it is time for it to engage the target. It also is possible (by the use of additional algorithms) to represent the command and control aspects of a network.

MISSION PREPARATION

Because a training mission with a dense environment will involve a large number of objects to be simulated, it is appropriate to define as much as possible about the mission environment prior to the actual training. This leaves the instructor free to oversee and supervise the training rather than being burdened by the details of defining and updating the threat environment. Normally, however, the instructor may take control of threats defined in the environment, or he may delete threats or add new threats during the training session. This allows him to modify the environment to meet any special needs as might occur from one training session to another.

Prior to the training, the mission definition effort includes defining threat types and their locations or paths which will used in the mission. This data is stored, typically, in a "mission" data file. This file then can be used repeatedly with various students for a given training session. A typical trainer would probably have dozens of already prepared mission files available for use from day to day. New missions would be prepared as new training needs are uncovered. Old missions would be updated from time to time to maintain an up-to-date training capability and to adapt to changing training needs.

Much detailed data is needed to describe the threat environment for a mission. Each signal of a threat emitter requires, perhaps, dozens of parameters (frequencies, pulse rates, antenna characteristics, etc.) to be defined. A typical emitter will have several signals, and will exhibit several modes of operation (each with different signal characteristics). Each threat platform also has a number of parameters characterizing its capabilities such as accelerations, speed, turn rate, climb/dive rate, and altitude limitations. The result is that it is not practical for an instructor to simply define all of the data needed to describe each threat to define a mission. It is much more practical to use libraries of data to simplify the mission definition effort.

LIBRARIES

A typical trainer (such the A-6/F-14 Trainers, for example) utilizes data libraries which contain the parametric data needed to describe the details of a threat, its emitters and signals, its platforms and weapons, and so forth. As the instructor defines a mission (usually using an off-line mission preparation mode of the trainer) he designates a threat type and assigns it a location (if a fixed site) for example. The mission generation program then accesses data libraries to obtain the detail threat emitter signal data. The necessary data is then provided by the mission generation process to the mission file (from both the instructor inputs and from libraries) to completely define the parameters of the threat and its elements that will be later needed during a training session to accomplish the threat simulation.

Some of the data is fixed for a particular kind of threat. For example, the limitations of an aircraft platform (maximum altitude, speed, etc.) are the same for several different examples of the same type of threat. Other data has variations from one example to another. Threat emitter signals will vary in frequency and pulse repetition rates (at least a little) from one case to another. In such cases the threat library information might be in terms of ranges of each type of parameter. The mission generation program would then utilize that information to "pick" an exact parameter value for each threat case in the mission.

A trainer would typically have a number of libraries to support the mission generation process. These would include threat emitter libraries, signal libraries, platform libraries, weapon libraries, and threat reaction algorithm libraries, for example. It is also might be noted that some of the library data may not be stored in the mission files. An alternative is the use of library information available to the trainer during training. The mission file then might have simply the designation of the type of threat platform, weapon, or whatever, the instructor defined during mission definition. During training the real time simulation software would access the defined library data to obtain the complete details needed to accomplish the designated threat simulation. The A-6/F-14 Trainers, for example, utilizes both library approaches, i.e., during mission preparation some data is taken from libraries and put in the mission file, and other data is obtained from other libraries during the training mission.

It can be seen that the use of libraries greatly reduces the effort required for an instructor to define a mission during mission preparation. Much of the detailed threat data is basically repetitive and therefore lend themselves to a library approach. Some of the training mission threat definitions are mission peculiar, however. A simple example is the location (latitude and longitude) of a fixed position ground threat. Another is the flight path for an airborne threat for a particular mission. Such mission peculiar information must, of course, be defined by the instructor for each mission. Normally it possible, however, to "cut and paste" and otherwise borrow and edit information from other missions to help in defining a new mission.

Libraries may also provide advantages in updating training. For example, if new information becomes available about a particular threat's signal parameters, the corresponding portion of the threat library might be first updated, and then the library used to reprocess and update all training missions. Care must be used in the organization of the library implementation to insure that this is possible, however. Mission correction via. library updating is therefore a factor which should be considered in the mission preparation and library system design approach selection.

CONCLUSIONS

Modern trainers can simulate the effects dense threat environments allowing realistic training to be accomplished in this critically important area. Recent events have certainly confirmed the value of having the equipment, tactics and training to counter a dense threat environment as well as demonstrating the penalties of lacking those capabilities. It is to be expected that an even increased emphasis will be placed on these capabilities for trainers designed in the future.

About the Authors:

Mr. Hunter is a Senior Design Analyst in the Training And Simulation Division of AAI Corporation. He has worked in the simulation field for 13 years and is currently the lead software engineer for the threat tactics software for the A-6/F-14 Aircrew Trainers Suite. Mr. Hunter obtained a BS in Mathematics from Frostburg State University and a MS in Computer Science from The Johns Hopkins University.

Mr. Puckett received a B.S. in Electrical Engineering from The Johns Hopkins University and has worked at AAI in the design of training and simulation devices, particularly Electronic Warfare training systems, for more than 20 years.

A ROBOTIC SYSTEM CONCEPT FOR PARTIALLY AUTOMATING THE SECOND ECHELON OPPOSING FORCE AT THE NATIONAL TRAINING CENTER

Admiral S. Piper
U.S. Army Project Manager for Training Devices (PM TRADE), Orlando, Florida

William F. Kraetz
Alliant Techsystems, Edina, MN

ABSTRACT

The paper reports the results of feasibility study investigating the potential of applying robotics to partially automate the second echelon opposing force (OPFOR) at the National Training Center (NTC) at Fort Irwin, California. A general robotic system concept approach is developed in the context of a generic unmanned robotic vehicle model. Training Wheels, a particular implementation of the generic robotic system concept is overviewed. The Training Wheels system concept consists of a command post,manned by 1 or 2 persons, for monitoring the movement of multiple vehicle company convoys composed of multiple unmanned vehicles following the path set by a lead vehicle manned by 2 persons. The concept appears to be technically feasible as it makes effective use of key operational constraints, operator personnel, and a supervised autonomous control schema for controlling the unmanned convoy vehicles.

INTRODUCTION

This paper reports the results of a recent study performed to determine the feasibility of applying unmanned vehicle technology (robotics) at the US Army's premier combined arms training facility, the National Training Center (NTC) at Ft. Irwin, CA. The goal of the effort was to effectively represent (via robotics) a Warsaw Pact style second echelon force with significantly fewer operator personnel.

Following the introductory section, the paper is organized in four sections. First, the background for and requirement for the study problem is discussed. Second, five unmanned vehicle technology concepts are introduced and a generic robotic system model is defined to provide a framework for the discussion of the Training Wheels concept in the next section. Here the Training Wheels concept is introduced/overviewed and characterized in terms of the generic robotic system model defined in the previous section. In the final section conclusions from the study are summarized.

BACKGROUND/REQUIREMENT

In January 1987, the Chief of Staff of the Army approved a plan to "ramp" to brigade operations at NTC. The plan called for a phased approach beginning with the evaluation of the brigade support "slice", allowing limited brigade operations with two maneuver battalions in the force-on-force engagement exercise.

Initially, the plan called for a third maneuver battalion, to be added eventually, to bring NTC up to "full" brigade operation. But, as a result of changing mission needs and costs constraints it was recently decided not to add a third heavy maneuver battalion. The need for additional OPFOR is still considered valid since the current OPFOR cannot portray doctrinal force ratios for some missions against the two battalion/task force brigade.

The senior Army leadership recognized that full BLUEFOR brigade operations would necessitate an increase in the size of the opposing force (OPFOR) from a motorized rifle regiment (MRR) to a motorized rifle Division (MRD) in order to maintain doctrinal force ratios. Figure 1 depicts how these units would be distributed on the NTC Battlefield in terms of the first and second echelon areas of operation. The direct approach for achieving the increase in the OPFOR, e.g. total replication of men and equipment was proposed and deemed to be unaffordable. As an alternative, it was decided to energize industry to look for technological solutions (e.g. artificial intelligence (AI), and robotics) that would allow the OPFOR to grow to an MRD without a linear increase in personnel and equipment.

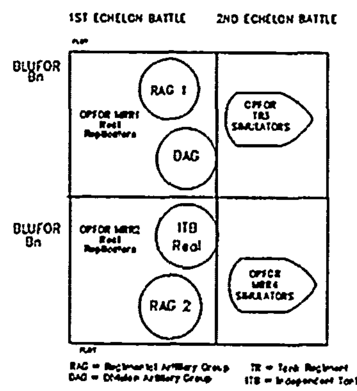


Figure 1. NTC Expansion Concept

Subsequently, in May 1987 a NTC Industry Day was held at Ft. Irwin. The following February (1988) PM TRADE issued a Broad Agency Announcement (BAA) seeking innovative technological solutions to several NTC specific problem areas. In the BAA, the OPFOR problem was characterized in terms of first and second echelon components which have distinctly different operational requirements, as summarized in Table 1, suggests two different technological approaches, i.e., manned vehicle/reduced crew technology (first echelon), and unmanned vehicle/robotic technology (second echelon).

First Echelon:	*Second Echelon:*
Overall veh/crew performance should not be either enhanced or degraded significantly.	Min manpower required (<1) man/vehicle.
Veh/crew must be capable of engaging multiple moving & stationary targets out to 3km.	Convoy column mobility
Veh/crew's ability to execute evasive maneuvers cannot be compromised.	Instantaneous path selection for real time responsiveness.
Role of tank cdr in selecting & prioritizing targets must be maintained.	Human presence in column req'd.
Safety cannot be compromised.	Evasive maneuver capability: speed up & vary interveh space.
	Veh speed: 20-30 km/hr on rd. 10 km/hr max off rd
	Inter veh spacing. 50-100 m variable.
	MILES detection capability.
	Interface convoy to CIS.

Table 1. Key OPFOR Operational Requirements/Constraints

In 1989, separate study contracts for first echelon (manned vehicle technology), and second echelon (unmanned vehicle technology) were awarded. The results of the first echelon feasibility study are described in (1), and (2) and will not be discussed further here.

The rest of this paper will provide an overview of the second echelon (unmanned vehicle technology) feasibility study performed by Alliant Techsystems.* A detail technical report (3) describing the Training Wheels concept is available to government agencies through the Defense Technical Information Center.

CONCEPT DEVELOPMENT

The second echelon robotic system concept developed (as will be seen in the next section) was guided by the five broad unmanned vehicle concepts highlighted in the 1988 NTC BAA and summarized in Table 2. Examination of Table 2 reveals all concepts have a "man" presence, some more and some less. In addition, all concepts result in systems that can be less manpower intensive than the "full" crew implementation.

(1) Autonomously operating vehicles occasionally monitored & controlled by mobile and/or fixed based operator and/or operator teams.

(2) One operator controlling at least a squad-sized element.

(3) One operator controlling at least a platoon-sized elements.

(4) A two-man operator team controlling squad and/or platoon-sized elements.

(5) A three or more man operator team controlling company sized elements.

Table 2. Unmanned Vehicle Concepts

The process of selecting a concept is iterative and involves always keeping the goal in view while considering the system issues summarized in Table 3, and the key second echelon requirements and constrained summarized in Table 1. Results from this analysis clearly leads to concept 5, "a three or more man operator team controlling company sized elements", in Table 2.

This concept can be characterized at some high level of abstraction as a system composed of three major subsystems, i.e., manned and unmanned subsystems connected by a communication link, see figure 2. The components of the model are briefly described next.

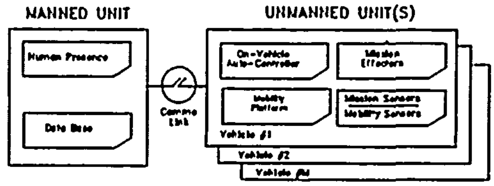


Figure 2. Generic Unmanned Robotic Vehicle System Model

ISSUES

- o The number of unmanned vehicles to be controlled.
- o Distribution of the Control Task (man vs machine).
- o Intelligence/automation that can be cost effectively embedded on the unmanned vehicle.
- o Type of training scenarios to be executed.

Table 3. System Issues

The major functions included in the unmanned subsystem are (1) on-vehicle auto-controller, (2) mobility and mission sensors, (3) mission effectors, and (4) mobility platform. These functions working collectively and in concert with "off" vehicle functions permit the unmanned vehicle to interact with and affect the environment to accomplish the mission. The manned subsystem is characterized by a manned presence and data bases to augment the man functions. The manned presence maybe distributed over both fixed and mobile platforms, and will in general form a command, supervise, and/or operate control schema of the unmanned subsystem(s). Communication between the manned and unmanned subsystems is essential. The characteristics of the link are a function of control schema, the operational environment, and the particular application. This model is quite general and provides a convenient tool for framing the discussion in the remainder of this section and the next section where the Training Wheels concept is introduced.

For the selected robotic system concept to be viable, it must support the efficient and effective execution of at least the key system functions shown in Table 4. These functions are discussed next in the context of the functional components of the generic robotic system model shown in Figure 2.

---

* In July 1989, the feasibility study contract was awarded to Honeywell Advanced Systems Center (prime) who teamed with Kaman Sciences Corporation (Sub). In October 1990, the Advanced Systems Center became part of Alliant Techsystems.

o Command and Control

o Maneuver

o Execution of Mission Specific Functions

Table 4. Key System Functions

Command and control of the unmanned robotic vehicle system is effected through the on-vehicle auto-controller, mobility and mission sensors, and the reliable exchange of data between the unmanned and manned subsystems. Effective execution of this function is essential for safe and precise movement of the unmanned robotic vehicle, and operation of mission equipment.

Maneuvering the unmanned robotic vehicle is accomplished by the mobility platform responding to heading and speed commands from the on-vehicle auto-controller. The on-vehicle auto-controller receives guidance and instruction from the lead vehicle via the communication link. Safe and effective maneuvering of the unmanned robotic vehicle is affected by terrain, accurate position location, route planning and selection, and driving.

Execution of mission specific functions are effected through mission sensors, effectors, and the on-vehicle auto-controller all operating directly or indirectly on instructions from the human via the communication link.

The difficulty of performing these functions varies greatly across application domain. It is clear that there are at lease two keys to success, human presence, and the amount of autonomy obtained by the unmanned robotic vehicle. The OPFOR second echelon training application is an exemplar of this as we shall see in the next section.

TRAINING WHEELS CONCEPT

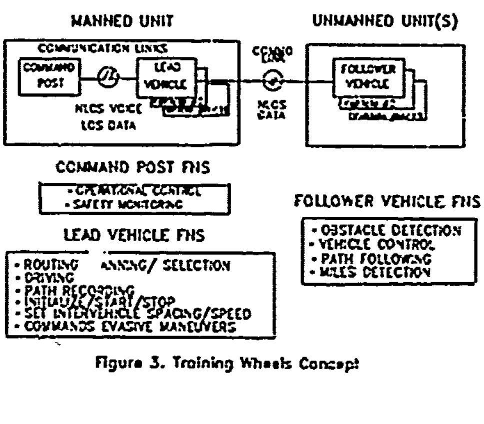


Figure 3. Training Wheels Concept

The Training Wheels system concept developed by Alliant Techsystems, see Figure 3 above, resulted from: a broad knowledge of robotics technology and its application; a understanding of how the NTC functions and the Army trains there; an understanding of the unique NTC environment; and a careful examination and understanding of the training and operational requirements. Specifically, the five key operational constraints, summarized in Table 5, were recognized and exploited to essentially "crack" the problem and lead to the potentially affordable and effective solution.

o Very restricted operational domain.

o Listed engagement requirement.

o Restricted mobility (column formation, albeit, arbitrary paths).

o Friendly RF environment (albeit restricted).

o Obstacle detection for safety only, met navigation.

Table 5. Key Constraints

As can be inferred from Figure 3, the Training Wheel concept, is slight variation of the Concept presented in the last section. The three major subsystem structure remains with the following exceptions.

The Manned subsystem, characterized by a distributed man presence, consists of the three components: a command post (CP), manned by no more than two personnel, provides the command and control function. The CP will be physically located in the Core Instrumentation System (CIS) facility at NTC. The CP is connected to the lead vehicles through the second component, a global communication network which includes LOS data and NLOS voice links. The third component, the manned lead vehicle has, a crew of two, a driver and column operator. The driver selects the path and speed, and pilots the lead vehicle transmitting its path as series of way points over a local data network. The column operator monitors the unmanned follower vehicles status and controls intervehicle spacing.

The Unmanned subsystem consists of a unmanned follower vehicles which uses way points transmitted from the lead vehicle to navigate and execute accurate track-in-track vehicle following.* Since the follower vehicle accurately follows the path of the lead vehicle, only simple (relatively speaking) obstacle detection is needed.** The path is known to be clear except possibly for transient obstacles such as humans and animal crossing the column path after the lead vehicle passed. The obstacle detection system need only detect and report these types of problems to the column operator.

* In October 1990, USALABCOM's Team Program (Robotics) demonstrated an accurate (±18 inches) path retrace capability for a robotic vehicle at the AMC Technology show held at Aberdeen Proving Ground, Maryland.

**In June 1990, Alliant Techsystems demonstrated the viability of a obstacle detection system in a desert environment. Both vehicle and personnel type obstacles were tested.

Although there are hardware and software implementation issues remaining it seems reasonably to say the Training Wheels concept is capable of effectively executing the unmanned robotic vehicle essential functions discussed in the previous section. It achieves them through the innovative placement of operator personnel and the unique semi-autonomous vehicular navigation and control schema.

CONCLUSIONS

The Training Wheels concept appears to be technical feasible to accomplish the training mission envisioned at the NTC. The concept uses a form of supervised autonomy to control columns of 10 to 15 unmanned vehicles from a manned lead vehicle, thus achieving significant manpower reduction. This system performance relies upon several key operating constraints: very restricted operational domain; limited engagement requirement; restricted mobility (column formation); obstacle detection for safety not for navigation.

REFERENCES

[1] aum, David R. et al, "MANPRINT approach to AI/ROBOTICS for NTC OPFOR augmentation", final technical report (31 August 1990), Hughes Simulation Systems, Inc., Training & Control Systems Division, Advanced Systems Facility, Minneapolis, MN.

[2] Baum, David R. et al., "Reduced crew for opposing force surrogate vehicles at the National Training Center", Proceedings, Interservice/Industry Training Systems Conference November 6-8 1990, pp. 110-122.

[3] Kraetz, W.F., "Application of Robotic Convoy Technology to Combat Training Centers: final report (Aug 1991), Alliant Tech Systems Group, Edina, MN.

SUGGESTED READING

Kraetz, W.F. and Krantz, D.G., "Design of a Realistic Feedback System for Controlling a Teleoperational Land Vehicle", Proceedings, 1989 Association of Unmanned Vehicle Technical Symposium, July 1989.

Kraetz, W.F. and Krantz, D.G., "Training Wheels: Supervised Autonomous Vehicles for NTC OPFOR Expansion", Proceedings, 1990 Great Lakes Chapter, AUVS Technical Symposium, 7 Nov 1990.

Kraetz, W.F. and Krantz, D.G., "Supervised Autonomous Vehicles for NTC OPFOR Expansion", Proceedings, NATO Seminar on ROBOTICS in the Air-Land Battlefield, March 1991.

ACKNOWLEDGMENTS

The authors are indebted to the highly talented individuals, too numerous to list here, who composed the Government, Alliant Techsystems, Honeywell, and Kaman Sciences Technical teams performing on this effort. We especially want to acknowledge Vickie McDevitt, Sarah Gottschalk, Arlene Gregory and Aurelia Vallin for their collective contribution in typing the early drafts and final preparation of the paper for publication.

ABOUT THE AUTHORS

Mr. Admiral Piper is a Research Electronics Engineer within the Engineering Management of Instrumentation/Tactical Engagement Simulation Branch at PM TRADE in Orlando, Florida. Since 1985, he has been PM TRADE's technical focal point for training device technology applications of AI and Robotics. He directed both first and second echelon feasibility studies for application at NTC. He holds MSC Degrees in Electrical Engineering, Engineering Mechanics, and Mathematics from the University of Alabama-Tuscaloosa, and is a member of IEEE, Eta Kappa Nu, and Sigma Xi.

Mr. Kraetz served as the principal investigator for the Training Wheels program with PM-TRADE to utilize robotic technology for the cost-effective insertion of a second echelon at the NTC.

With Alliant Techsystems (Honeywell) Ordnance Division for over 10 years, Mr. Kraetz is currently Technical Director of the Teleoperational Vehicles Area leading development efforts for reconnaissance and forward observation applications. He has been involved with numerous military systems in a lead engineer/technical director capacity, most notably the Howitzer Improvement Program (HIP), the Modular Azimuth Positioning System (MAPS), and as IR&D manager of Honeywell's Robotic Vehicle Test Bed program.

Mr. Kraetz holds a B.S. Electrical Engineering and Computer Science degree from Marquette University with specialties in computer engineering and digital control systems.

A HIERARCHICAL RULE-BASED ARCHITECTURE FOR IMPLEMENTING INTELLIGENT ADVERSARIES IN A SIMNET ENVIRONMENT

Avelino J. Gonzalez	Dan Mullally	Gilbert Gonzalez
Computer Engineering Dept	UCF - IST	UCF and IST
University of Central Florida	Orlando, FL.	Orlando, FL

ABSTRACT

The availability of intelligent adversaries in a training simulator environment can clearly enhance the training experience for students. However, implementation of this capability into simulators has been slow as well as difficult. The semi-automated forces presently available for SIMNET, although quite sophisticated, still represents a partial solution, as the name itself indicates.

Representation of tactical expertise in rules gives rise to the problem of encapsulating every possible scenario within simple rules. This could lead to the need for a very large number of rules which, not only would have to be developed, but would also have to be efficiently executed in a real-time environment. This represents an unacceptable situation.

Improvements could be made by grouping rules according to the mission being simulated, but the number of rules required would still be large, and there would be no benefit of reusing situational knowledge commonly required in different missions.

The approach described in this paper is to develop a hierarchical ordering of rules which, at the highest levels, can be used to recognize the general situation being faced by the adversary. Examples of these situations are when an adversary needs to remain hidden from the student, or when it is appropriate to attack the student. Recognition of this high level situation will activate a lower level set of rules which will attempt to implement the prescribed course of action within the context of the situation. These will, in turn, activate another set of rules which will carry out the low level implementation details of the action within the simulation software.

1.0 INTRODUCTION

It is clear to anyone that in a networked simulated training environment such as SIMNET, the endowment of intelligence to a simulated representation of enemy forces represents an advantage in terms of tactical training. Making the correct tactical decision depends not only on evaluating the "static" situation such as the terrain, the weather, and the mission, but also on what the enemy does. Therefore, it is advantageous that the enemy entity behave in an manner representative of that of an actual enemy's.

AI techniques address the issue of representing and modelling human intellectual behavior in specific circumstances. The best developed sub-field in AI is that of knowledge-based systems. Using deductive reasoning as well as other means, but without actually modelling the brain, knowledge-based systems attempt to simulate the heuristic decision-making process followed by people knowledgeable in a specific domain when faced with a problem in that domain.

Since the enemy forces can be presumed to be knowledgeable in the tasks for which they have been trained (i.e., tactical warfare), it is only natural that knowledge-based systems be considered a prime candidate for use in this task. Nevertheless, there exist some significant obstacles to the use of knowledge-based systems for the task at hand:

The first one of these is that the knowledge possessed by a battle entity (i.e., an attack helicopter, a foot soldier, a tank platoon etc.), contains a large measure of common sense as well as survival instinct. Knowledge-based systems are not particularly suited to representing common sense or instinctive behavior. Although a certain amount of

these can be adequately represented, it is virtually impossible to represent all of the common sense knowledge accumulated by individuals throughout a lifetime, or their in-born instinct to survive in a dangerous situation.

Secondly, knowledge-based systems are best employed when a limited set of inputs (i.e., scenarios) initiate the exercise of a base of knowledge in order to choose which of a limited set of alternatives to implement. A battle situation, however, can present to the student a virtually endless range of scenarios to which he has to properly react. To prescribe a course of action for each of the possible scenarios which could be presented is not a realistic means of implementing a knowledge-based system. That would require an immense base of knowledge which would be nearly impossible to capture and manipulate effectively as well as efficiently.

Thirdly, knowledge-based systems are typically inefficient and rather slow. This has been partly due to the traditional use of symbolic languages by researchers in the technical community. Such languages as LISP or PROLOG are generally interpreted, memory-intensive, and built for flexibility rather than speed. The more recent trend, however, is to develop knowledge-based system tools in the conventional languages such as C or ADA, thus somewhat alleviating the problem. Nevertheless, knowledge-based systems remain a comparatively slow means of computing when compared to conventional algorithmic methods.

The research described in this paper first analyzes the knowledge to be used by the intelligent opponent in the SIMNET environment and proposes an architecture for implementing it.

2.0 THE NATURE OF TACTICAL KNOWLEDGE

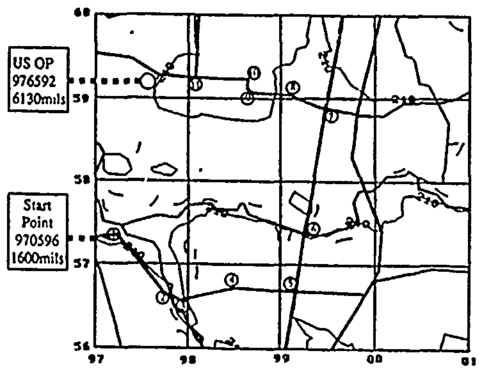
In order to devise an architecture to implement intelligent adversaries, it became important to understand his mode of thinking. It was decided to elicit such knowledge from an experienced tank commander, using the SIMNET environment as a demonstration testbed for that knowledge. The vehicle of interest was a single T-72 main battle tank (for the Warsaw Pact forces). The domain was limited to a reconnaissance mission along a road with the objective of searching for the enemy (which in this case was actually the blue forces). Upon reaching the final destination set out in the mission, the simulated vehicle was to take up a concealed point of observation and report all enemy activity until it was either attacked, discovered, or the main column of friendly forces reached its location.

2.1 Knowledge Elicitation Methodology

The methodology used was a combination of the traditional question-and-answer interview and observation. In the q&a phase of the process, the focus was on learning about military terminology and procedures, as well as some basic tactics. The capabilities of the weapon systems were also discussed as they might affect the tactics. This was carried out intermittently along with the observation phase.

The observation phase had two objectives: one was to observe the expert's tactical decision-making process, and two, to understand the capabilities of the SIMNET environment in which the simulated intelligent adversary would be located. This phase was composed of carrying out a simulated reconnaissance mission on a specifically-chosen piece of terrain in SIMNET in order to observe the expert's tactics. The terrain chosen was a relatively flat region around Ft. Knox, KY, which is represented in the SIMNET terrain data base. Figure 1 depicts the region used. An enemy tank (a "Blue force" tank in this case) was positioned near the destination of the mission in order to determine the range of visibility possible in SIMNET.

Semi-Automated Forces
Route Recon



CP#1 973574
CP#2 97
CP#3
CP#4
CP#5
CP#6
CP#7
CP#8
CP#9
CP#10
CP#11

FIGURE 1

2.1.1 Observation About SIMNET

There were a number of significant observations made about SIMNET, which although not directly related to the objectives of this exercise, nevertheless provided a valuable insight into SIMNET and its effect on an intelligent opposing force simulation. Some of the more significant ones are as follows:

- Treelines, although very realistic from a distance, have no depth. Thus, they only provide concealment from the enemy when it is on the other side. However, it is difficult to position the tank to hide behind it and still be able to see from behind it. In order to do the latter, the tank has to protrude far enough through the treeline that it loses concealment.
- The tank crew is permanently limited to a closed-hatch condition, which does not allow dismounting in order to look over the top of a hillmass before exposing the entire tank. This limits the tactics that can be employed.
- Forest canopies are basically circular treelines. In other words, once inside the forest, the entire inside of it can be seen immediately. This is also not realistic, in that threats which could in reality could be hidden in the depth of the forest, can be seen immediately upon entering.
- It is somewhat difficult to ascertain, even from relatively close, whether a line is a road or a river. Additionally, bridges are defined only as the intersection of a road and a river. There is no superstructure or abutment to indicate a bridge from a distance.
- Lastly, there was considerable flicker in the horizon. This may be a symptom of the particular tank being used, and not a generic problem with SIMNET, but it could have been a reason why one of the authors incurred simulator sickness during the observation exercises.

2.1.2 Observations About Tactical Knowledge

Although the exercise on SIMNET provided a good initial base of knowledge, it's relative flatness and deforestation did not provide the appropriate environment for testing some of the tactics learned.

This led the investigators to create an imaginary terrain region on paper which included the type of features which would provide for more challenging tactical decisions. Figure 2 shows that terrain. The observation process now shifted from a SIMNET exercise to one of pencil-and-paper. This presented the advantage that terrain features could be altered in order to more deeply investigate the fine points of tactical maneuvering. Additionally, in order to understand the appearance of tanks in a more varied terrain as well as to understand the firing capabilities of the tanks in question, several exercises were held with the TOP-GUN gunner training simulator.

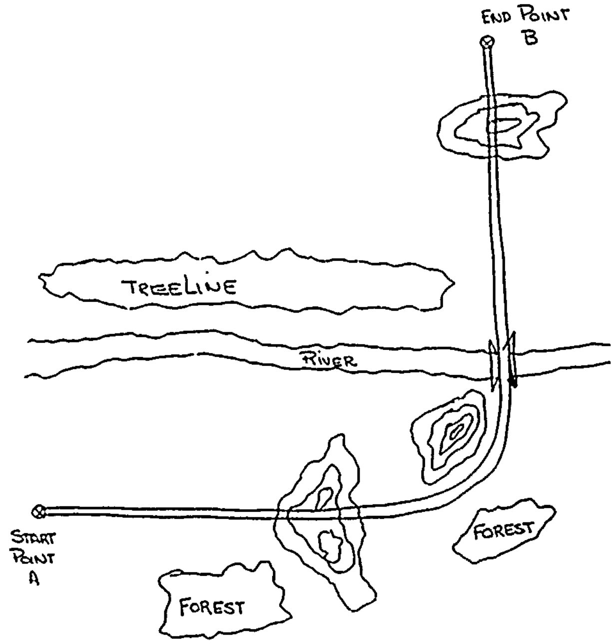


FIGURE 2

This process resulted in the following bits of knowledge:

- A reconnaissance mission requires tactical movement so as not to be discovered, and/or attacked by enemy forces which may be in the area. This is in contrast to administrative movement, which implies travelling somewhat more leisurely down a road, without the expectation of enemy contact.
- Tactical movement is composed of tactical path planning, a sub-task which requires the selection of the next immediate destination that will serve as a steppingstone to the final destination, and which will best provide protection from vulnerability to enemy sighting or fire.

2.2 Tactical Path Planning

Tactical path planning, as a sub-task to tactical movement, will be chosen based on:

- Mission: The type of mission will dictate whether tactical movement or administrative movement is required; whether the enemy should be engaged

or whether retreat is preferred. In a reconnaissance mission, the objective is to see and not be seen. Therefore, engaging the enemy is prohibited unless acting in self-defense. *Reconnaissance-by-force,* on the other hand, attempts to find the enemy by presenting a target and drawing fire in order to engage the enemy and mislead him into thinking that it is part of the main force. Other missions which were not investigated are *attack, hasty attack,* and *retreat* as well as others.

- Terrain: The terrain features which are being included as part of the investigation are hillmasses, treelines, forests, rivers, roads and bridges. Each of these have different features as they affect tactical path planning. For example, hillmasses provide cover from direct ground fire. This is the most desirable terrain in which to be for protection against ground-based enemy. Treelines provide concealment from ground-based elements, but provides no cover since fire can typically penetrate the treeline. Forests can provide cover as well as concealment, depending on their size and orientation. Generally, they are not to be entered unless they are sparse. Rivers typically represent obstacles to movement. In this investigation, they are all assumed to be unfordable, and therefore, crossed only over bridges.

- Enemy Presence: The key issues are the direction of enemy threat, and the likelihood of having contact with the enemy. The levels of possible contact are *definite, likely,* and *unlikely.* The actual presence of the enemy is represented as a simple boolean yes or no.

2.3 Sequence of Events in Tactical Path Planning

In the process of trying to move tactically through an unfriendly territory in a reconnaissance mission, a tank crew commander will typically go through the following sequence of events in making a tactical path plan:

- Situational awareness, composed of:
  - terrain appreciation
  - weather
  - damage assessment (if any)
  - remaining stores (fuel, ammo,etc)
  - enemy assessment
- Course of action selection (tactical knowledge)
- Course of action implementation

Situation awareness knowledge is that which is used by the commander to review the situation and recognize the important points which he will use in determining the optimal course of action. Some of the things which this includes are recognition of potential danger (i.e., an ambush), being under fire, completion of mission, obstacles to completion of mission, avenues of approach, vulnerability to enemy sighting and fire, recognize concealment, cover, chokepoints, canalization, going too far out of the way and many other features about the terrain.

Tactical knowledge is then what the commander's experience indicates should be done in order to carry out the overall objective of the mission. This should include knowledge on tactical movement, conduct of fire, reaction to threat, mission execution, multi-vehicle tactics, escape or hasty retreat, and attacking, among others.

The course of action implementation should include physically moving the simulated entity from one place to another, avoiding minor obstacles such as a tree, realistic acceleration and deceleration, stop and turn the vehicle, fire, etc.

A further discussion of these events as they would affect the implementation of a semi-automated force will be included in the next section.

2.4 Rules of Tactical Path Planning

Rules of tactical movement are an example of the tactical knowledge discussed in the previous section. These were developed as applicable to a route reconnaissance mission, with an objective of getting from a starting point A to a final destination Z, establishing a point of contact at the latter, and reporting any enemy activity detected. The following are some rules which were extracted from the domain expert during the investigation. A later section describes the actual implementation of these rules.

1) Seek the closest cover from present position that leads closer to final destination.

2) Seek concealment when close-by cover in the general direction of the final destination is not available.

3) Move rapidly when cover and concealment are not available.

4) Select next partial destination before leaving a covered or concealed location.

5) Minimize time of vulnerability to enemy fire or sighting.

6) If time is of the essence, sacrifice cover and concealment for speed of movement.

7) Unmask potential locations of enemy before proceeding to them if possible.

8) Unmasked terrain becomes masked once again if unobserved for longer than thirty minutes.

9) Go to nearest cover when there exists concealment which is only slightly closer than said cover, and both are in the general direction of the destination.

10) Do not proceed to terrain elements which are further than 150 meters away from the roadway.

11) If enemy contact has been made and the enemy has seen you, seek cover as soon as possible, report their position and change mission to reconnaissance by force.

12) If enemy contact has been made and he has not seen you, then seek concealment, report sighting, maintain contact, and remain out of sight.

3.0 AN ARCHITECTURE FOR THE IMPLEMENTATION OF AN INTELLIGENT ADVERSARY IN A SIMNET

In a semi-automated force environment, a mission determination task could be called ultra-high level knowledge and it could be represented by a human, depending on the type of tactics to be taught, and the terrain to be employed. Although this could also conceivably be done with a knowledge-based system, it was clearly beyond the scope of this investigation.

Tactical knowledge, on the other hand, represents the essence of what a tank commander does in a tactical path planning task. This is a highly heuristic task, which is most clearly applicable to knowledge-based systems.

The situation awareness task is intermediate in nature because tactical path planning requires an assessment of the situation before a course of action can be chosen. It is probably best represented by a combination of knowledge-based and algorithmic techniques. The algorithmic portion would be composed of mathematical functions which can describe the lay of the land, the location of significant terrain features, and the distances between them and the tank. The heuristic parts would be the classification of the various features into a context usable by the higher level knowledge, for example, that a hillmass provides cover, but that it is too far away to be of immediate consideration.

The implementation of the course of action is the lowest level routine described here. It represents the actual carrying out of the selected course of action by the tactical knowledge module. This would best be carried out algorithmically since its function is rather procedural in nature. Figure 3 is a graphical description of the above levels.

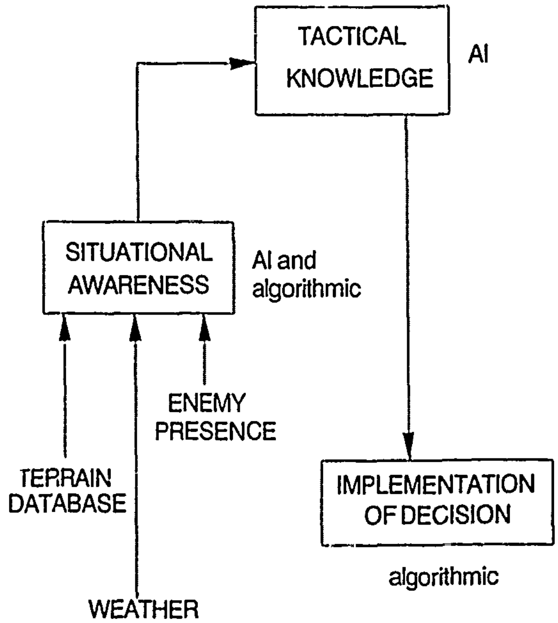


Figure 3 - High Level Knowledge Representation

3.1 Knowledge Representation Paradigm and Generalized Techniques

It quickly became clear to the investigators that the knowledge possessed by the expert was generally that of an If-Then rules. This was strongly hinted in the course of previous sections.

Although rules are a powerful means of representing and exercising knowledge, the need for frames became obvious just as quickly. Frames would allow for a richer representation environment where all the data calculated by mathematical functions, or new inferences made by the inference engine would be stored. The frames' capability for demons would be highly useful in calculating distances to various terrain features as well as determining the rate of usage of fuel, and the remaining store of ammunitions. Inheritance, on the other hand, could facilitate the representation of knowledge about a tank or other battlefield entity.

It is important, however, that an innovative technique known as *hierarchical rule classification* be implemented. Without this, representation of the problem through rules would be infeasible due to the numerous individual scenarios which would have to be identified and the rules to respond to each of these written explicitly.

3.2 Hierarchical Rule Classification

The concept of this technique is that behavior can be generalized such that rules can be written that would have applicability in a number of different scenarios. This could circumvent the need to write a rule for each possible different scenario, which would be clearly unworkable. The rules written in section 2.4 are general in nature, so that their implementation would be consistent with this approach.

The significance of this is that various sub-levels of knowledge would be required in order to classify the situation and provide inputs so that the generalized rules could use the information to make a determination. The knowledge used to carry out the latter would also be generalized except to a lesser degree, but it would also need lower levels of knowledge to support its task. This hierarchical relationship of more generalized, higher-level knowledge supported by more specific lower levels of knowledge is referred to as hierarchical knowledge representation, and represents an innovative technique to use in the solution of the intelligent adversary knowledge representation and execution.

For example, in the context of a route recon mission that is the subject of this investigation, a highest level rule would say that

> If the mission is route recon, and enemy contact is likely,
> Then tactical movement is recommended.

Another rule which would perform a similar function at the same level would be:

> If the mission is route recon and enemy contact is unlikely,
> Then administrative movement is recommended.

Assuming that tactical movement is recommended, the presence of that fact would activate another chunk of knowledge which would be used to assess the situation. It would have to interface with data present in some attached databases such as the terrain database, as well as use some of the givens of the problem such as weather conditions, the state of the stores, the damage incurred, etc. This chunk would in fact be the situation awareness module and serve to support the tactical knowledge chunk which would be the next one to be activated. For example,

> If the distance to treeline A from the present location is the shortest to the tank's present destination of all other treelines,
> Then treeline A is the closest treeline to tank.

Similarly,

> If the distance to hillmass B from the present location is the shortest to the tank's present destination of all the other hillmasses,
> Then hillmass B is the closet hillmass to tank.

Examples of other rules which support the tactical knowledge are:

> If hillmass B is closer to the final destination than the tanks present location
> Then hillmass B is in the correct general direction of the mission goal.

> If a terrain element is a hillmass,
> Then it provides cover.

> If the terrain element is a river,
> Then it provides an obstacle to movement.

> If the terrain element is a treeline,
> Then it provides concealment.

The tactical knowledge would then use this supporting information as follows:

> If the recommendation is tactical movement, and terrain element X is the closest cover, and X is in the right direction,
> Then proceed to terrain element X.

After the tactical knowledge chunk makes a decision as to where to move to next, then a lower level function would implement that decision. But assuming that there is contact with the enemy prior to the movement being made, and that fact is reflected by being under enemy fire, then another general rule would activate and it would look as follows:

> If under enemy fire,
> Then retreat to safety.

Another chunk that interprets a safe retreat would be activated next which would survey the situation again in a different light and recommend a new location to go. The block diagram in Figure 4 shows the hierarchical nature of the rules shown above.

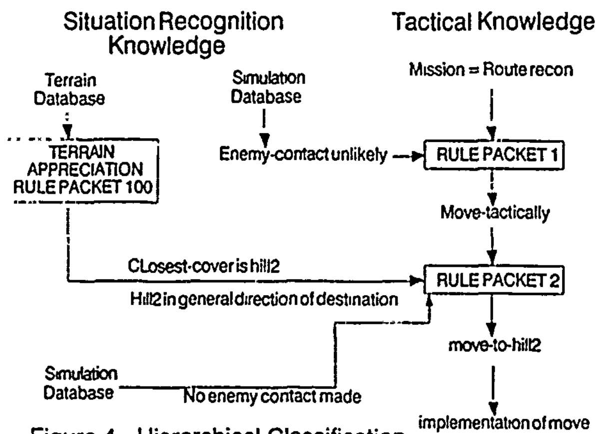


Figure 4 - Hierarchical Classification

4.0 PROTOTYPE DEVELOPMENT

The last stage of this investigation was to put into practice all the techniques discussed above as a prototype which would demonstrate their feasibility.

The objective of the prototype was to have one opposing force tank autonomously perform a route recon mission as described in the previous chapter.

The first decision faced was to decide what tool to employ in its design.

4.1 Knowledge-based System Tool Selection

Upon consideration of various commercially-available tools which could be used to develop knowledge-based systems, it was decided that CLIPS represented the best compromise between sophistication, performance and price. Developed by NASA Johnson Space Center, CLIPS is a PC-based, rule-based shell which employs the Rete algorithm in a pattern-matching environment. A distant relative of OPS-5 and ART, it is quite powerful, and relatively easy to use. Additionally, due to the fact that it is written in the C language, its interfacing with the C-based SIMNET system testbed being developed at IST would be infinitely simplified. The only weakness seen was that version 4.3 does not support frames at the present. However, it is understood by the authors that the upcoming version 4.4 will have that capability.

4.2 Initial Prototype Description

In order to familiarize the research team with the CLIPS tool as well as be able to better understand the knowledge being extracted, a simple initial prototype was designed which would conceivably act as an on-bard assistant to an inexperienced tank commander in a route recon mission.

This prototype is a stand-alone, and it acquires information by asking the commander about the mission and the nearby terrain features. Although simple in nature and admittedly unrealistic in its assumptions of commander attention, it nevertheless performed its intended function well and provided a starting point for the further development of the final prototype.

4.3 Final Prototype Architecture Recommendation

The final prototype was designed to be implemented directly into the SAFOR testbed being developed at IST. It would require no interaction with the tank commander and would direct other routines to move the tank to a new location. It would be usable under any terrain or set of conditions that would be applicable to a route recon mission. It would interface indirectly with the SIMNET terrain database so as to be as realistic as possible for its intended SIMNET environment. Although not completed at the conclusion of the author's summer assignment, its design will be described below as a recommendation for further research in the topic of intelligent adversaries in a simulated training environment. Please note that due to the lack of time to complete its implementation, the design described is by necessity, a very high level one with little detail having been defined.

The basic architectural design of the advanced prototype would be in a object-oriented environment (C++), where the tank(s) in the simulation would be an instantiation of a tank object with such attributes as its present location, straight-line distance to the final destination, number of rounds remaining, speed capabilities, river fording capability, amount of fuel left, damage assessment, and various others. The tank object would be able to call methods which would assist it in its tactical path planning. These methods would be embedded CLIPS packages which could, in turn, access other auxiliary C functions in order to carry out its task. Figure 5 shows a graphical description of this concept.

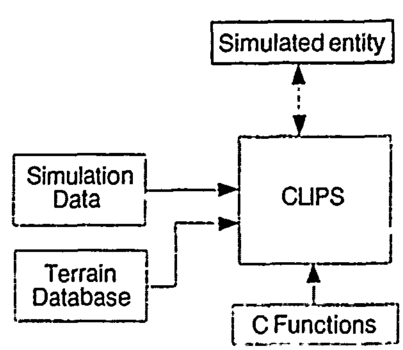


Figure 5 - General Architecture of Advanced Prototype

Many questions and details need to be filled in in this design. The largest remaining question at this point is the memory and processing time requirements of this type of arrangement on the limited resources of the simulation computer. It is hoped that the implementation of the high level prototype design described above would help answer these questions.

4.4 Future Research

Besides completing the advanced prototype described in section 3.3 above, there are other areas which merit investigation into its applicability to the problem at hand.

For example, an object-oriented extension to C such as C++ could conceivably simplify as well as enhance the some of the functionality of the testbed. Object-oriented extensions are ideally suited to simulations of independent objects which behave differently from other objects in the same environment. It also allows for easy modification and additions at a later date. And, although not yet proven, it is likely that the use of C++ will not introduce any additional overhead into the computations.

Other alternatives which could be employed if CLIPS proved ineffective in this application would be:

- a blackboard architecture system such as the Generic Blackboard. This has some definite drawbacks, but also great potential.
- a distributed processing environment.

Additionally, the knowledge for missions other than route reconnaissance would need to be extracted. Although not likely, this could reveal a significant complication in the endowing of intelligence to opposing forces.

5.0 SUMMARY AND CONCLUSIONS

In summary, it can be safely said that AI techniques such as knowledge-based systems are quite applicable to the problem of intelligent adversaries. In fact, it is believed by the authors that truly intelligent behavior in a simulation object cannot be practically achieved in any other way. The research performed under this grant was successful in achieving the following:

- analysis of the problem
- extraction of a limited portion of the tactical knowledge
- preparation of a scenario to use in testing the prototypCCe
- selection of a knowledge-based tool
- selection of a knowledge representation paradigm
- development of a simple prototype
- preliminary design of an advanced prototype system

12TH I/ITSC-1990: SIMNET FIGHTER AIRCRAFT APPLICATION

Captain Brian K. Rogers
Clarence W. Stephens
Armstrong Laboratory, Human Resources Directorate
Aircrew Training Research Division
Williams AFB, AZ

Alan B. Oatman
Bolt, Beranek, and Newman Inc., Systems and Technologies Division
Williams AFB, AZ

ABSTRACT

This paper describes the preliminary investigation defining problems of expanding realtime simulation of fighter aircraft to a distributed simulation network. The 12th Interservice/Industry Training Systems Conference was selected as the test site to prove the concept. I/ITSC provided an arena for linking simulators from several manufacturers and laboratories. Six fighter aircraft simulators participated in the network along with a simulated ground control intercept (GCI) station and semi-automated forces providing threat aircraft.

I. INTRODUCTION

Networking Background

The Defense Advanced Research Projects Agency (DARPA) initiated development of technology for a credible force-on-force simulation for the Army. This program has yielded significant breakthroughs in realtime large force, ground battle simulation. This technology has now transitioned from research and development programs to being incorporated into the Army's training programs.

These large force simulations offer numerous advantages over field exercises. Lack of adequate exercise locations for every Army unit often meant lengthy periods between large force exercises. Other factors included in the cost of actual field exercises; transportation, vehicle maintenance, fuel, and ammunition limit the ability to practice in large force, live fire exercises with decreasing defense budgets. To overcome these difficulties and reduce costs, simulation exercises were believed to be able to complement unit training. However, the ability to network large numbers required for realistic training needed to be developed. Simulation networking (SIMNET) has been developed under DARPA's guidance. Numerous types of ground vehicles and aircraft (generally helicopters) have been simulated and networked together. Now hundreds of devices at several Army installations can conduct networked war games.

The Air Force Human Resources Laboratory, Aircrew Training Research Division, now reorganized as the Armstrong Laboratory, Human Resources Directorate,

Aircrew Training Research Division (AL/HRA) has taken the initiative to apply SIMNET concepts to fighter aircraft training. AL/HRA has pursued the ability to network aircraft simulations to provide team training for aircrews and their controllers. Our approach thus far has been to maintain SIMNET compatibility. Maintaining SIMNET compatibility is achieved by keeping all the existing information within the version of SIMNET the Army is using and adding extra information required to operate aircraft avionics. . Ensuring compatibility with existing SIMNET will lead to the ability to conduct large scale joint simulation exercises. The AL/HRA effort will develop the technology to connect aircraft simulations into the Army's SIMNET.

Network Participants

For the 12th I/ITSC, 5-8 November 1990, Orlando, Florida, up to six fighter aircraft simulators and a ground control intercept (GCI) station were networked using a modified SIMNET protocol. Simulators participating in the demonstration were developed by three different manufacturers and AL/HRA's and AL/HRG's (Logistics Research Division) on-site contractors. Other network fighter aircraft participating in the 12th I/ITSC SIMNET application included; 2 AL/HRA F-16 Combat Engagement Trainers (CET) one in the Air Force Human Resources Laboratory booth the other at Williams AFB, AZ, an F-16C Air Intercept Trainer (AIT) at Williams AFB and an AL/HRA F-15 Modular Aircrew Simulator System (MASS), developed by McDonnell Douglas Training Systems, located in the Paragon booth, a General Dynamics A-16 simulation in their booth, and an F-16 part task trainer (PTT) connected from Fort Rucker, Alabama. All network participants are shown in Figure 1. Other applicable network participants included the semi-automated forces (SAF) from the Bolt, Beranek and Newman (BBN) booth generating threat aircraft and ground forces and a GCI simulator in the Air Force Human Resources Laboratory booth. The GCI simulation was developed and manned by AL/HRG, Wright-Patterson AFB, OH.

AITs were originally designed for the Air Force National Guard and Reserve to serve as a hands-on, throttle-and-stick (HOTAS) trainer for F-16s. AIT usage has expanded to include all F-16 replacement training units for the initial acquisition of radar/weapons HOTAS skills. AITs consist of basically a three cubic foot box, containing the VME chassis, side mounting plates for the throttle and stick and an opening for a chair as the seat. Radar information is presented on a radar multifunction display (MFD) built into the chassis front and the out-the-window and heads-up-display (HUD) is provided by a 19" Silicon Graphics monitor, placed on top of the chassis, driven by a Silicon Graphics 4D/70GT Personal IRIS operating at 60 Hz non-interlaced.

CETs are also part-task trainers for F-16Cs. CETs were developed at AL/HRA to research the feasibility of a glass-cockpit providing the fidelity required to train radar and tactical employment skills as a follow-on to the AIT. A CET consists of; a cockpit shell with side-stick, throttle, seat, 25" head-down touch screen display for cockpit instrumentation-radar MFD, radar

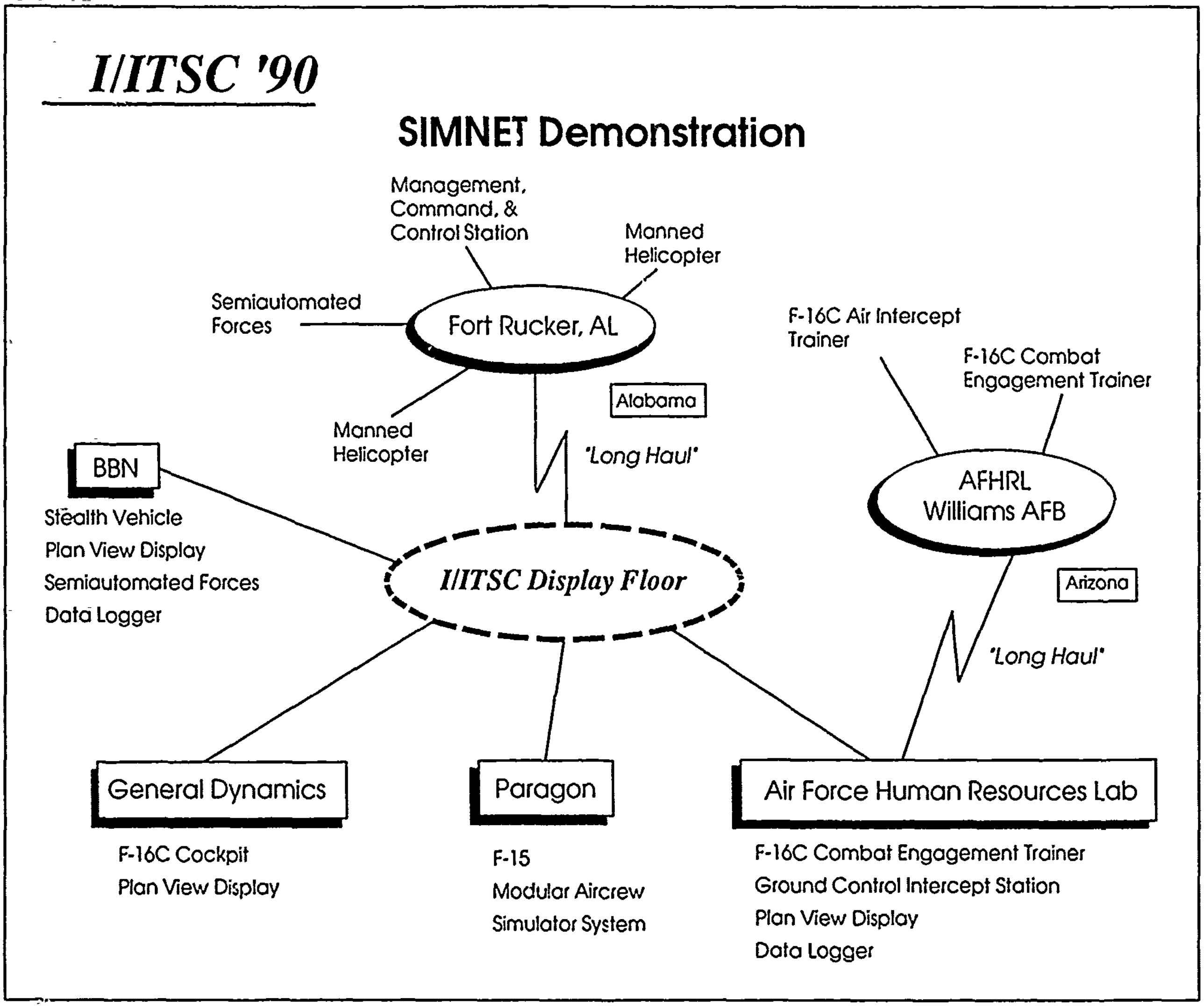


Figure 1 Network Configuration-12th IITSC, 5-8 November 1990

warning, stores management, and flight instruments all driven by a VME-based simulation, and a single 19" Silicon Graphics IRIS display for a limited out-the-window and HUD generated by a Silicon Graphics 4D/70GT Personal IRIS. The head-down display has been covered by a molded plastic overlay to show only the instruments. Touch screen response is achieved through tactile feel buttons built into the overlay. In the SIMNET configuration the network interface unit (NIU) is added directly into the VME bus.

Audio capability for the CETs is available using the UHF radio system, transmitting via the push-to-talk switch on the throttle assembly. Audio output is directed from the CET to the SIMVAD portion of the NIU where it's digitized and passed via SIMNET voice protocol onto the Ethernet network. Audio reception from

other network participants is received and processed based on the radio packet information. The NIU acts as the filter for receiving voice traffic from the network. If the NIU is set to the same exercise number and radio frequency, accomplished during initialization, the NIU will process and pass the digitized packets to the SIMVAD subsystem for conversion into an audio signal. The signals are then passed to the CET audio system and on to the pilot's headset.

Audio capability for the AL/HRA AIT and F-15 MASS simulation, F-16 PTT at Fort Rucker, and AL/HRG's GCI simulator consisted of interface units with push-to-talk switches connected directly to the NIU's SIMVAD subsystem. The General Dynamics A-16 audio system was interfaced to the simulation similarly to the CETs.

The F-15 MASS was integrated to the NIU as a cooperative effort of BBN and McDonnell Douglas Training Systems personnel. Initial integration of the F-15 MASS simulation to an NIU was accomplished at AL/HRA, final integration was accomplished in Orlando prior to the 12th I/ITSC opening.

The General Dynamics A-16 integration to the NIU was entirely an internal company project. The NIU was loaned by AL/HRA for their integration work and use through the 12th I/ITSC. The NIU and SIMNET protocol definition documentation were provided to General Dynamics by BBN at the request of AL/HRA. This cooperation expanded the user base for applying SIMNET to an industry developed, fighter aircraft simulation. General Dynamics gained the ability to connect into the distributed simulation network for threats and cooperative aircraft for joint tactics employment. AL/HRA and BBN benefitted by having another user providing inputs for future upgrades to the SIMNET protocol.

The F-16 PTT connected from Fort Rucker was a generic fixed wing simulation developed by BBN. The simulation was a modified AH-64 simulator, an A-10 HUD, a GAU-8 cannon, and BBN flight dynamics for an F-16A. Eight channels of out-the-window imagery was generated by a BBN Advanced Simulation, GT 111 image generator. Seven channels are generated out to 3.5 km with the eighth channel-area of interest, boresighted to the centerline, generated to 7 km. All image generation, aircraft performance, and sensors were generated by the GT 111. A situational display, god's-eye view was available in lieu of simulating a radar.

The GCI station from AL/HRG was developed under contract by System Research Laboratory. The GCI station can simulate either a 407L Tactical Air Control Station (TACS) or an E-3A Airborne Warning and Control Station (AWACS) display. The GCI station simulation was hosted on a SUN SPARC IPC station linked via Ethernet to an NIU. The GCI station simulation operates in a receive only mode. The NIU captures only aircraft messages then passes them to the SUN system. During the 12th I/ITSC the GCI simulated an AWACS display. The only data passed from the GCI NIU to the network is when a radio transmission is made from the GCI controller. Radio calls from another simulator to the GCI controller were handled the same as for the AIT.

II. MISSION SCENARIOS FLOWN

During the course of the 12th I/ITSC several overall mission scenarios were flown. The one scenario which remained constant throughout was the General Dynamics A-16 and F-16 PTT mission. The A-16 flew only close air support (CAS) missions teamed with the F-16 PTT long-haul connected from Fort Rucker. The A-16 was attacking targets generated and controlled by the BBN SAF systems. The missions the AL/HRA simulators flew varied depending on the simulators connected to the network and whether the long-haul connection to Williams AFB was active. Only air-to-air (A-A) missions were conducted due to AL/HRA's simulation capabilities. The SIMNET protocol was certainly capable of supporting air-to-ground (A-G) operations as evidenced by the General Dynamics A-16 and F-16 PTT mission.

In scenarios involving the I/ITSC network and the long-haul to Fort Rucker, SAF generated air threats-MIGs. These scenarios would be modified to allow both offensive counter-air (OCA) and defensive counter-air (DCA) missions by the CET and the F-15 MASS simulators. SAF would generate MIGs in either a point defense combat air patrol (CAP) or in a sweep toward the CET and F-15 MASS in their defensive CAPs. SAF generated MIGs were basically dumb targets with no weapons, flying programmed routes.

For the times when the SAF MIGs were flying a CAP, the CET and F-15 MASS functioned in an OCA mission role. They would perform sweeps to engage and destroy the SAF MIGs using their own radars and GCI guidance. During the other missions the CET and F-15 MASS were in a point defense role defending against the attacking SAF MIGs.

During the twelve periods when the long-haul connection with Williams AFB was active, the link with the main I/ITSC network was severed due to the data limitations of the 56kbps (kilo bits per second) data line. Prior to the conference we estimated the network capacity for a 56kbps line would only accommodate the data transfer for six aircraft simulators. Since the I/ITSC network involved three aircraft plus the one from Fort Rucker and up to eight MIGs generated by SAF (SAF v. 3.9.12 could generate 60 vehicles total), the decision to have a separate network configuration when connected to Williams AFB was made prior to the conference. The decision to use a 56kbps line was made on the basis of cost. If a T-1 line bandwidth had been available, the need for a separate network configuration for the long-haul connection would not have been necessary.

In the long-haul configuration the CET and AIT at Williams AFB were a team in an OCA role against the CET and F-15 MASS with GCI control at the I/ITSC conference. Digitized voice radio communication was available between the AIT-CET and CET-MASS-GCI. The GCI controller provided real-time updates for the CET and F-15 MASS at I/ITSC. The AIT-CET pilots were required to acquire and appropriately target and employ against the CET and F-15 MASS threat.

Typically A-A missions lasted 5-10 minutes. The A-16 and F-16 PTT predominantly continued to employ in the CAS role throughout the conference and were generally not reset. Scenario

participants were initialized using the SIMNET Plan View Display (PVD). Altitude, airspeed, and heading were assigned during each set-up. Reinitialization between missions was accomplished in less than a minute. Normally the CETs, AIT, and F-15 MASS were the only cockpits being initialized between missions. SAF MIGs were reset for the SAF system at the BBN SIMNET control station.

III. SIMNET PROTOCOL

The protocol used for integrating the simulations was the DARPA/BBN SIMNET Release 6.6 protocol. Original plans called for the use of protocol extensions to demonstrate enhanced capability to pass active sensor data on the network. This was not completed prior to the 12th I/ITSC and therefore SIMNET Release 6.6 Protocol Data Units (PDUs) were implemented. This protocol allowed rapid integration of dissimilar devices, via the use of the Network Interface Unit developed by AL/HRA and BBN Systems and Technologies Division.

The Protocol Data Units supported in the NIU for the 12th I/ITSC were:

* Activate Request
* Activate Response
* Deactivate Request
* Vehicle Appearance
* Fire
* Impact

along with support for sending and receiving voice packets.

This is not an all-inclusive list of SIMNET packets, however it was adequate for the demonstration. These packets allowed the simulators involved to participate with a minimum of performance degradation. The simulators were initialized into the situation using the activate pair, were displayed across the network using the vehicle appearance packet, could launch air-to-air missiles with the fire packet and indicate the result of the launch with the impact packet, and finally, could tell the network they were leaving the situation using the deactivate packet. This level of network support was deemed sufficient without protocol extension for the purposes of demonstration.

Although protocol extensions were not included for the 12th I/ITSC demonstration, progress in this area has continued. Protocol extension is required to integrate a weapon's sensor systems onto a SIMNET network. When parameters of importance are found that don't fit into the existing protocol, an extension is necessary. The first area considered for protocol extension was radar transmissions. Although SIMNET 6.6 includes a radiate PDU, it was insufficient to support the radar warning sensors on the aircraft.

A new PDU, the radar PDU, has been implemented since the conference. The radar PDU was designed to support radar interaction between cockpits on the network without severely loading either entity involved in the relationship. This is achieved by including a list of the players within the radar beam. This list is not intended to be all-inclusive, but to allow the sharing of data to simplify the detection problem for players. By also specifying moding and directional parameters, remote

players can uniquely identify the type of energy transmitted. Use of the data transmitted in the PDU, in conjunction with data kept in a host table allows rebuilding of the radar volume if the receiving simulator requires such. The radar system and mode is communicated using a hierarchy to allow remote players to decode only those elements it can use. Very sophisticated hosts may use all the data presented, however, limited fidelity devices may interpret only the highest level of data.

Protocol extensions needed in several other areas have also been identified; although not implemented prior to the 12th I/ITSC. The principle areas of interest are in countermeasures and weapon guidance. The field of countermeasures includes both electronic defenses, as well as expendables for the purpose of defeating active sensors. Electronic countermeasures may be supported by the existing radar PDU, but further research is in progress to investigate this possibility. In the area of weapon guidance, consideration is being given to the ability for the launching vehicle to transfer fly-out responsibilities. This would allow the fly-out calculations to be performed by the device most capable, rather than forcing the launch vehicle to determine weapon termination.

Considerable research remains in the area of protocol extensions; especially in the electronic warfare area. The effort described here has not attempted to resolve all concerns, but to maximize the utility of the existing network for AL/HRA's research and development evaluation.

IV. VISUAL DATABASE ISSUES

To ensure an acceptable degree of visual correlation among the AIT, CETs, and the other networked aircraft simulators participating in the conference, two versions of the Hunter-Liggett visual database were developed by AL/HRA. An IRIS version was used for the AIT at Williams AFB and the CET on the I/ITSC conference floor. The IRIS provided each cockpit a single channel of visual display boresighted straight ahead of the aircraft centerline. The CET at Williams AFB required a second version of the database developed for the Advanced Visual Technology System (AVTS) for display on the Display for Advanced Research and Training (DART).

The AVTS supplied 5 channels of high fidelity imagery using a head tracking/channel switching algorithm to drive 7 display channels of the DART providing 300 degrees horizontal by 200 degrees vertical total field of regard. The CET/AVTS/DART, AIT/IRIS, and CET/IRIS hooked into the same SIMNET as the other conference participants providing a demonstration of networking dissimilar simulations and visual display systems.

The IRIS version was based on a BBN-supplied, Hunter-Liggett database in their SIMNET Database Interchange Specification format. Custom software was developed to transform the terrain polygons into an IRIS format. The IRIS terrain skin exactly correlated with the original BBN database. Due to limited development time, the BBN-supplied culture was not implemented in the IRIS Hunter-Liggett terrain database. Since the IRIS displays were used for A-A missions, the lack of cultural features was not noticeable. The correlation of the IRIS

database, without cultural features, to the other databases on different simulators proved to be completely adequate for A-A missions. Gouraud shading was implemented on IRIS databases providing a subjectively pleasing appearance. The general color of the terrain was modeled as green as opposed to BBN's original brown. IRIS moving models for other airborne players were displayed as F-16s.

The major problem encountered in the A-A scenarios was the limited size of the Hunter-Liggett database. The database was approximately 27 x 30 nautical miles.

The AVTS Hunter-Liggett database was based on Defense Mapping Agency, Digital Terrain Elevation Data (DTED) Level 1 and Digital Feature Analysis Data (DFAD) Level 1. The AVTS database modeling system is designed for DTED and DFAD input data. A software effort to implement the BBN terrain skin on the AVTS was considered and abandoned due to time constraints and image generator database incompatibilities. The terrain skin mismatch between the IRIS and AVTS databases was very slight with virtually no visible difference. Lack of visible differences is primarily due to the difficulty in discerning fine terrain detail displayed on the small IRIS screen. During the conference, no network information concerning other aircraft was passed to AVTS. Consequently no visual representations of other aircraft were available to the CET pilot flying at Williams AFB. This enhancement was implemented after the conference. The CET/AVTS pilot had to rely on radar contacts and audio communication to both the GCI and the wingman for air-to-air information.

The AL/HRA visual database effort in support of this conference was implemented in less than three months. The databases which were developed were adequate, although a bit small, for the A-A missions which were flown. The database effort also contributed greatly to the success of demonstrating networked simulations of air-to-air scenarios using SIMNET protocols with dissimilar cockpits and image generation systems.

V. DISCUSSION

Prior to this application of the SIMNET protocol, numerous concerns regarding the ability to network fighter aircraft using distributed simulation had surfaced. Some believed the dead reckoning of models and the large changes in position associated with fighter aircraft would cause several undesirable side effects which would render SIMNET useless in the high 'G,' rapidly changing A-A environment. Some of those concerns include; different CPU speeds (simulator, visual, NIU), weapon effects and fly-outs, and the ability to integrate an NIU to the simulation host.

The greatest concerns were related to the ability to network simulators of different CPU speeds. In the CET applications the VME is operating at 30 Hz, the IRIS 4D/70GT is at 18 Hz, and the NIU linked to the SIMNET Ethernet is at 15 HZ. In the limited scenarios using an A-A missile as the only weapon, no noticeable problems were encountered. No missiles missed due to slow position updates from the network. Visual jitter, aircraft jumping, was occasionally observed but did not preclude being able to fly in visual formation with other aircraft. Jitter was most apparent

during maneuvers of four G's or more. Maneuvers at lesser G's were virtually free of jitter.

To overcome any adverse effects on weapons fly-outs and their effects, the aircraft simulator shooting the missile continues to fly it in to the target. This allows the missile to fly as the shooter "sees" the target. This arrangement worked well for the 12th I/ITSC application when no countermeasures were available. Countermeasures-chaff, flares, and jamming will significantly affect weapon fly-outs. These countermeasures coupled with a highly maneuvering target aircraft will be much more susceptible to transport delays.

Target aircraft maneuvering and employing countermeasures faster than updates can reach the shooting-controlling simulator-aircraft would be counterproductive. If the transport delays in the network simulation system preclude the shooting aircraft from flying out a weapon, an alternative would have the target aircraft simulator fly-in the weapon. This would bypass the transport delays of having the shooting aircraft simulator fly-out the missile and possibly negate the effects of "last-minute" maneuvers and other countermeasures. If the target aircraft simulator were to takeover the weapon fly-in, after launch and activation (self guiding), all aircraft maneuvers and countermeasures would be included in modeling the fly-in. Any last minute maneuvers or countermeasures would be included in determining whether any damage occurred.

VI. CONCLUSION

SIMNET works for fighter aircraft simulators and will be implemented in the MULTIRAD program at AL/HRA to network 2 F-15 MASS, 2 CETs, several AITs, and a common threat simulation system. The MULTIRAD program will continue researching team training concepts and extend the capabilities of networked simulation. The first applications will be to provide a research testbed for beyond visual range (BVR), A-A engagements. As less costly and more capable visual systems become available, application to A-G missions for networked simulators will be made.

SIMNET offers an immediately available method of linking ground and aircraft simulators for large force-on-force simulations. By maintaining the basics of the SIMNET protocol the capability of connecting into the Army's existing network exists for truly integrated joint training exercises.

Incorporating full-up weapon systems trainers, within a TEMPEST area, into a realtime network allows practice with all aircraft avionics and employment of actual tactics along with all team members. These type exercises are not available now, short of actual combat. Team training will be the largest benefit of realtime, networked simulation.

ABOUT THE AUTHORS

Captain Brian K. Rogers is a Combat Systems Project Pilot at the Armstrong Laboratory, Aircrew Training Research Division. Capt. Rogers is developing networking standards to integrate a threat system for MULTIRAD. Capt. Rogers was an instructor pilot in EC-130H Compass Call and the T-37. Capt. Rogers possesses a Master's in Aeronautical Sciences from Embry-Riddle Aeronautical University and a Bachelor of Science in Aerospace Engineering from the University of Alabama.

Clarence W. Stephens is a Computer Programmer/Analyst at the Armstrong Laboratory, Aircrew Training Research Division. His responsibilities include directing the development of visual and sensor databases for studies and tests conducted by the division. Mr. Stephens holds a Bachelor of Science in Liberal Studies from the University of the State of New York.

Alan B. Oatman was the Lead Software Systems Engineer with Bolt, Beranek, and Newman's, Systems and Technologies Division, Williams AFB, Arizona. Mr. Oatman was the lead engineer on the installation and development of networking hardware and software for the MULTIRAD program. Mr. Oatman possesses a Bachelor of Science in Electrical Engineering from Washington State University.

AN OBJECTIVE LOOK AT THE

MODULARIZATION AND STANDARDIZATION OF TRAINING SYSTEMS

by
Gary M. Kamsickas
Boeing Defense and Space Group
Simulation and Training Systems
Huntsville, AL

ABSTRACT

Trends indicate, and projections suggest, that the future focus of training system design is modularization, reusability, standardization, cost reduction and team or multiple cockpit training. Several programs are in progress which deal directly with these issues. The Modular Simulator Design Program, a tri-service program administered by the United States Air Force, deals with the modularization and standardization of a single weapons system trainer. The Universal Threat Simulation System Program, administered by the United States Navy, is concentrating on the standardization and reusability of the threat and electronic warfare environment. Project 2851, administered by the United States Air Force, has the goal of standardizing radar and visual databases. In the team training environment, the Distributed Interactive Simulation program, administered by the United States Army, is attempting to provide a standard method of networking multiple training devices to allow for a cost effective team training environment. How these programs interact with each other is crucial to obtaining the goals of standardization, modularization, reusability and the eventual cost reduction of training devices. This paper provides an objective look at the interaction of these programs from a technical perspective. Suggestions are presented for possible modification to these standards to allow for greater compatibility.

INTRODUCTION

Over the past decade, in an effort to reduce costs, maintain technological superiority, and make use of emerging technologies, the government has attempted to standardize several facets of simulation and training technology. An early example of this effort was the attempt at standardization of software languages with the inception of the Ada programming language (MIL-STD-1815A) as the standard software language for all future training systems. Initially this standard was met with some reluctance on the part of industry. However, as the benefits of using Ada and the design methodologies associated with the language were gradually realized, industry acceptance of the standardization effort occurred. Standardization was perceived less as a "hand tying" effort on the part of the government and more as a benefit in both cost savings and advancing the state of technology for the entire simulation industry.

The government's effort to enforce the Ada programming language as a standard did more for the simulation community than provide code in a common language. It fostered the development of a new attitude in both the government and industry. This attitude was based on the Ada design principles of modularity, reusability, cost reduction, and also on just plain good engineering design practices. Design concepts and goals were expanded beyond modular, reusable, low cost software to the modularization and standardization of specific components of training systems, simulators as devices in general, and the networking of multiple simulation devices.

There are several major DoD research and development programs currently concentrating on developing standards which will significantly affect the training systems of the future. As with Ada, the standards associated with these programs are about to become enforceable requirements on future training systems.

It is the government's hope that enforcing these standards will result in a reduction in the cost and increase in quality of training devices through the reduction or elimination of redundant development efforts and a higher degree of system maintainability in a time of decreasing military budgets. However, without a conscious analysis of how these various standards may interact when invoked together for a training system, the result cannot be determined. A concern is that the result could be a training system which has been patched together in order to meet the requirements of all standards.

This paper discusses the programs associated with the standardization efforts and how these programs interact. Where appropriate, suggestions are presented for possible modifications to these programs/standards to allow for a better interface among the standards.

THE PROGRAMS

There are currently four major programs associated with the standardization effort; the Modular Simulator Design Program (MSDP), the Universal Threat Simulation System (UTSS), the Distributed Interactive Simulation (DIS) Program, and the Standard DoD Simulator Digital Data Base Common Transformation Program (Project 2851). Each program is producing a military standard and/or standard process for developing a specific technical aspect of the simulation arena. The following paragraphs provide a short synopsis of each program including program status.

Modular Simulator Design Program

The MSDP is a tri-service research and development program administered by the United States Air Force. The intent of this program is to develop a military standard for modular simulators which defines/standardizes module functions and module communication interfaces with respect to hardware, software and data. The goals of this program are to shorten simulator development schedules by promoting concurrent stand-alone module design, development, and test, reduce simulator costs through an increased competitive base and increased reusability, and improving supportability via well defined module functionality and interfaces coupled with recognized good design practices. This has been accomplished through the partitioning of a generic Weapons System Trainer (WST) into eleven distinct hardware and software modules as shown in Figure 1. Each of these modules communicates using a standard communication architecture comprised of a serial data bus (FDDI) using a standard set of system interfaces and communication rules. The Modular Simulator System (MSS) is defined by a set of System/Segment Specifications which define the system level requirements and the requirements for each module. These specifications along with a tailoring guide will eventually become the standard for modular simulators.

This program completed its demonstration/validation phase in December 1990 with successful results from tailoring the generic specifications to an F-16C application and building a working modular simulator. The standard is expected to be released by the government in the first quarter of 1992.

Universal Threat Simulation System

The UTSS program is another tri-service research and development initiative administered by the United States Navy. The intent of this program is to develop a threat generation system for threat models and threat data bases capable of providing training systems with a central repository of validated threat and interactive mode simulations. The goal of UTSS is to address the costs of controlling, maintaining currency and validating threat models and threat databases in simulators. To this end, the UTSS program will ultimately produce a Military Standard to address standard, reusable, current and

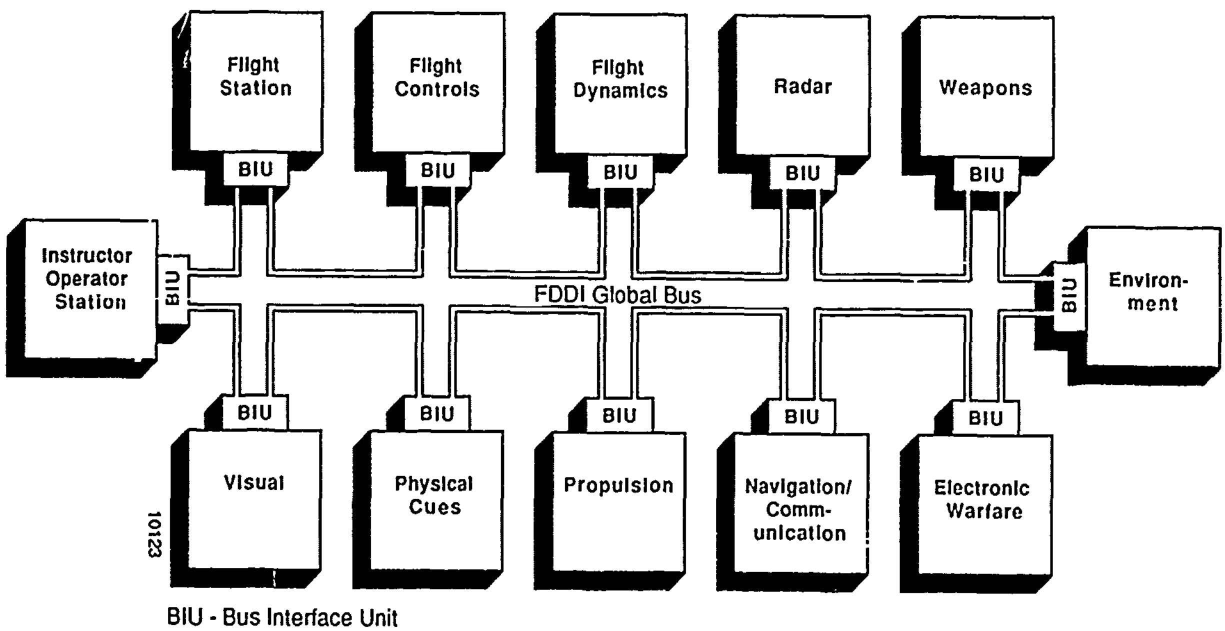


FIGURE 1
Modular Simulator Architecture

validated threat models and threat databases. In addition to the creation of the threat models and databases the UTSS will also provide a service or support facility responsible for the maintainence, update, validation and distribution of threat models and data bases to the simulator community. A concept for this support mechanism is shown in Figure 2.

The UTSS program recently completed its Front End Analysis (FEA) phase. The intent of the FEA was to define user training requirements and specify the level of threat fidelity to satisfy those requirements. These functional requirements will then be the basis of the UTSS performance specification and follow-on Full Scale Development effort. Preliminary results of the FEA indicated that users had identified a need for improved interactive threat models, current, accurate and validated threat data/models and an interface to the MSDP and DIS programs.

Distributed Interactive Simulation

The DIS program is a tri-service research and development program administered by the United States Army. This program addresses the standardization problems associated with interoperability among interconnected or networked simulators. This effort started in August of 1989 using the work of the Defense Advanced Research Projects Agency (DARPA) SIMulator NETwork (SIMNET) program as a baseline. The goal of this effort is to provide cost effective team training and developmental testing capabilities by using the current inventory of single operator trainers and future training systems. These trainers will be networked via Local Area Networks (LAN) and Wide Area Networks (WAN) as shown in Figure 3. In order to accomplish this goal the DIS standard must completely define the communication protocol between simulators in a distributed interactive simulation environment.

There is currently a draft Military Standard for the DIS. This standard identifies the DIS communication protocol as a set of International Organization for Standardization/Open Systems Interconnection (ISO/OSI) Application layer based Protocol Data Units (PDU). The PDUs are the communication messages for the DIS network. Each simulator (entity) participating in a DIS exercise is described to other participants via the PDUs. The DIS program currently has the basic PDUs defined to allow for entity interaction in a distributed environment. Interfaces such as network management, sophisticated electronic warfare, variations in fidelity among networked devices, and a common simulation environment remain undefined with the standard providing only basic guidance or recommendations in these areas.

Project 2851

Project 2851 is a tri-service research and development program which is also administered by the United States Air Force. The objectives of this program are to eliminate duplicative data base generation and redundant software development while improving database

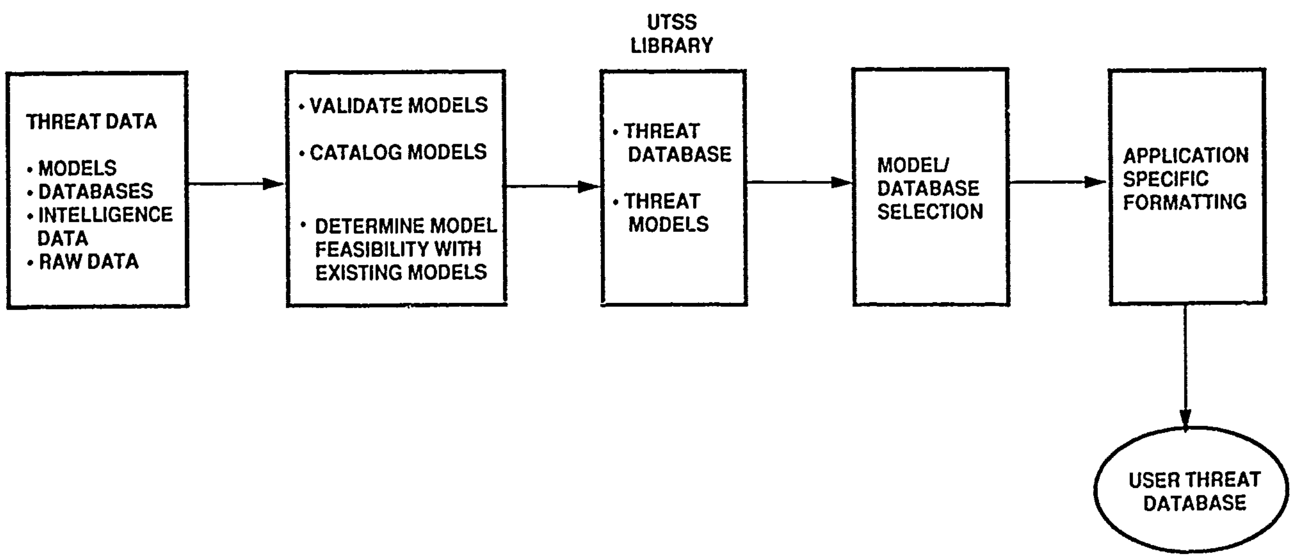


FIGURE 2
Universal Threat Simulation System Architecture

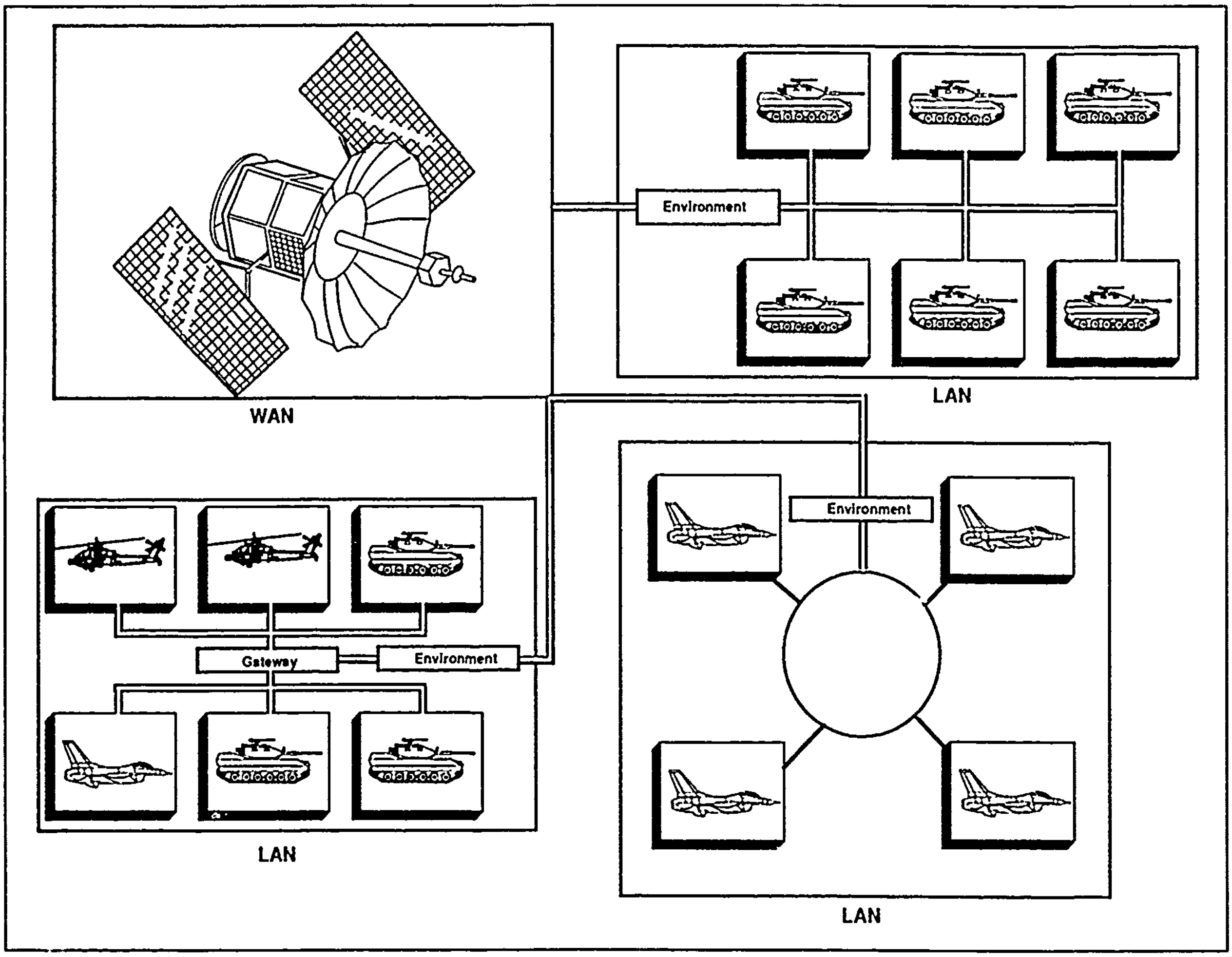


FIGURE 3
Distributed Interactive Simulation Architecture

consistency and correlation for visual and radar data bases. The end product of this program will be a data base production system/facility for visual and radar databases. This system will maintain Standard Simulator Data Bases (SSDB) for terrain, culture, models, and texture maps in the Project 2851 central libraries. The system will also provide a mechanism whereby externally created databases can be validated and entered into the SSDB (via the SSDB Interchange Format (SIF)). Generic Transformed Data Bases (GTDB, can be extracted from the SSDB for use in specific training system programs. Figure 4 provides a top level flow for the Project 2851 data base system.

This program is currently completing development of the Project 2851 data base production system. Draft military standards and handbooks for the SIF and GTDB have been produced and distributed to industry. The database system is expected to be operational in mid 1992.

THE GENERIC PROGRAM STRUCTURE

The government has executed each of these contracted research and development programs with the same basic structure. Although each program has tri-service support, one branch of the service is usually responsible for administering the program. The programs typically follow the schedule shown in Figure 5. Each program transitions through the same basic phases; concept development, concept design, demonstration/validation and finally standardization. The key element that all of these programs share is a heavy emphasis toward industry involvement in the standardization process. The government has in most cases mandated that the prime contractor for each program provide industry with a vehicle to provide constructive technical input into the design effort and the development of the standard. This is accomplished primarily through two methods, subcontracting of program effort and Industry/Service Working Groups (I/SWG). The latter, in the form of I/SWG meetings, are prevalent in all programs and allow industry a significant voice in the design/development of the associated standard.

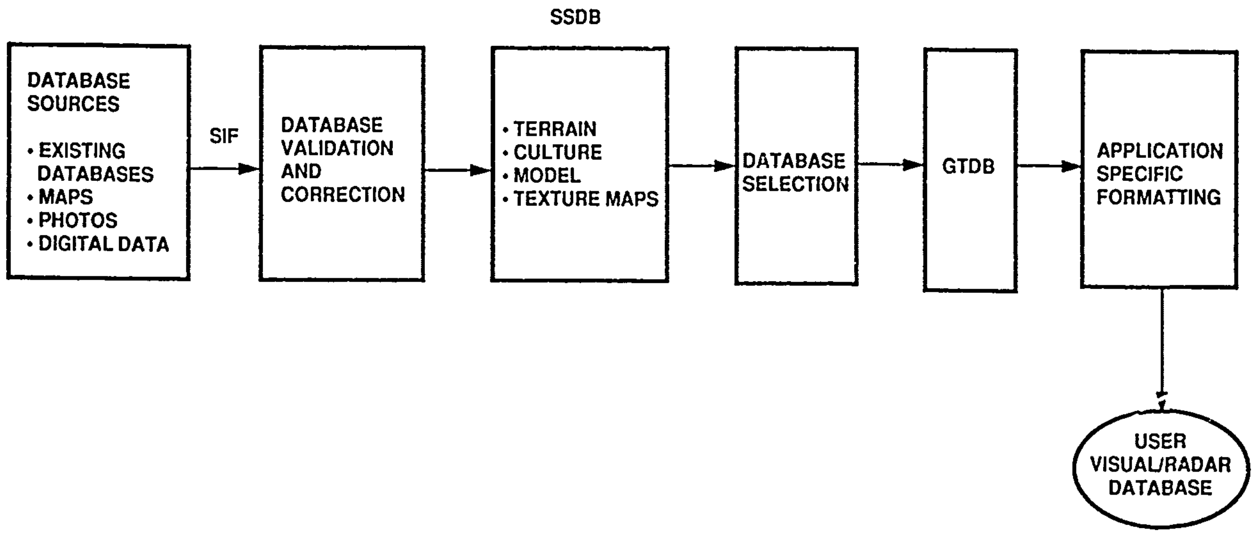


FIGURE 4
Project 2851 System Architecture

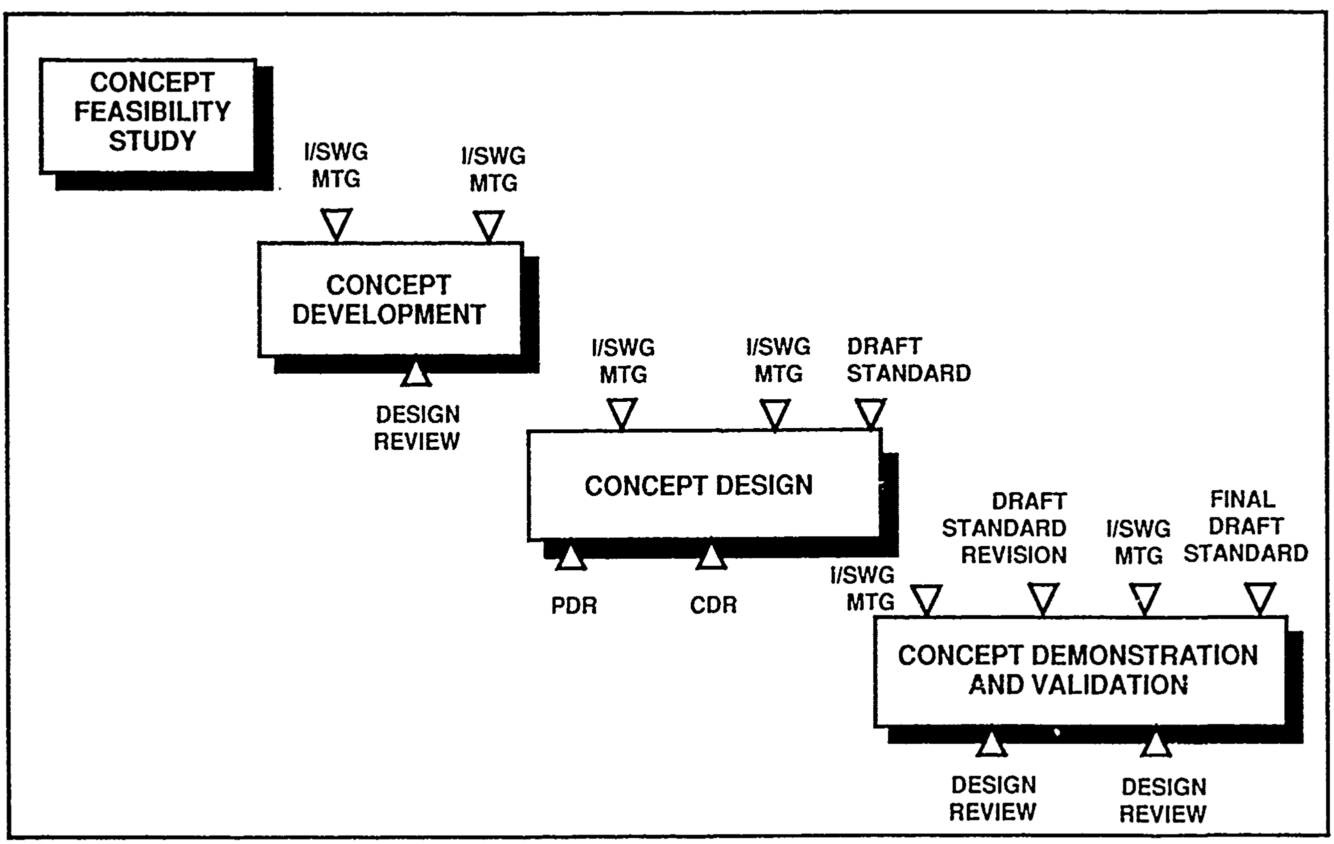


FIGURE 5
Generic Program Schedule

The I/SWG meeting concept has both positive and negative aspects. These aspects are shown in Figure 6. With the exception of the slow "design by committee" drawback associated with the I/SWG meetings, the remainder of the disadvantages are overcome by the positive aspects. With this in mind, one of the most difficult tasks of the I/SWG is to arrive at a consensus decision to resolve technical issues. Several methods have been employed including open debates/discussions, voting on issues, and the preparation of position papers. The best method is the requirement for a position paper to provide technical inputs for consideration. This accomplishes several objectives; 1) It forces the writer to clearly think through a technical issue instead of providing an off-the-cuff input, 2) It eliminates a great deal of redundant, and in most cases, circular discussion of an issue thus allowing for more effective and productive meetings, 3) The requirement to write a paper weeds out the people who are genuinely concerned about an issue from those who just want to debate, and 4) It allows for a paper trail of the inputs which caused the evolution of the design for future reference.

INTERACTION AMONG THE STANDARDIZATION PROGRAMS

One of the major problems with the standardization programs is the lack of coordination and interface between programs. Each program has its own requirements and in most cases is not required to investigate or comply with the other standard initiatives. This is due in most respects to the relative timing of the programs. As shown in Figure 7, the concept development phases and requirements analysis do not align among the programs. This causes difficulty in determining the impact of future, as yet undefined, standards on present standards initiatives except in a "prediction of the future" manner. For example, the impact of UTSS or DIS on the MSDP was difficult to assertain during the MSDP concept development and initial design phases. In most cases, each of the standards programs is aware of the other programs and make a fair attempt to consider them in the design. However, each program is usually in some state of flux. Therefore, when the scope or direction of one program changes all other programs must either realign or ignore the change.

I/SWG Advantages	I/SWG Disadvantages
• Industry provides input to the standard (get a vote in the creation)	• Design by committee is slow
• Industry experts provide experience to design process	• Some companies may attempt to bias design
• Differing viewpoints throughout industry avoids biased design	• Companies not paid to participate (some voices not heard)
• Industry and vendors aware of program direction and are better prepared with products when standard is enforced	• Companies attend to make an "appearance" (information gathering only)
• Foster a "Team" attitude between government and industry	• Company representative always changing (knowledge base and continuity of I/SWG always in a flux state)

FIGURE 6
I/SWG Comparison

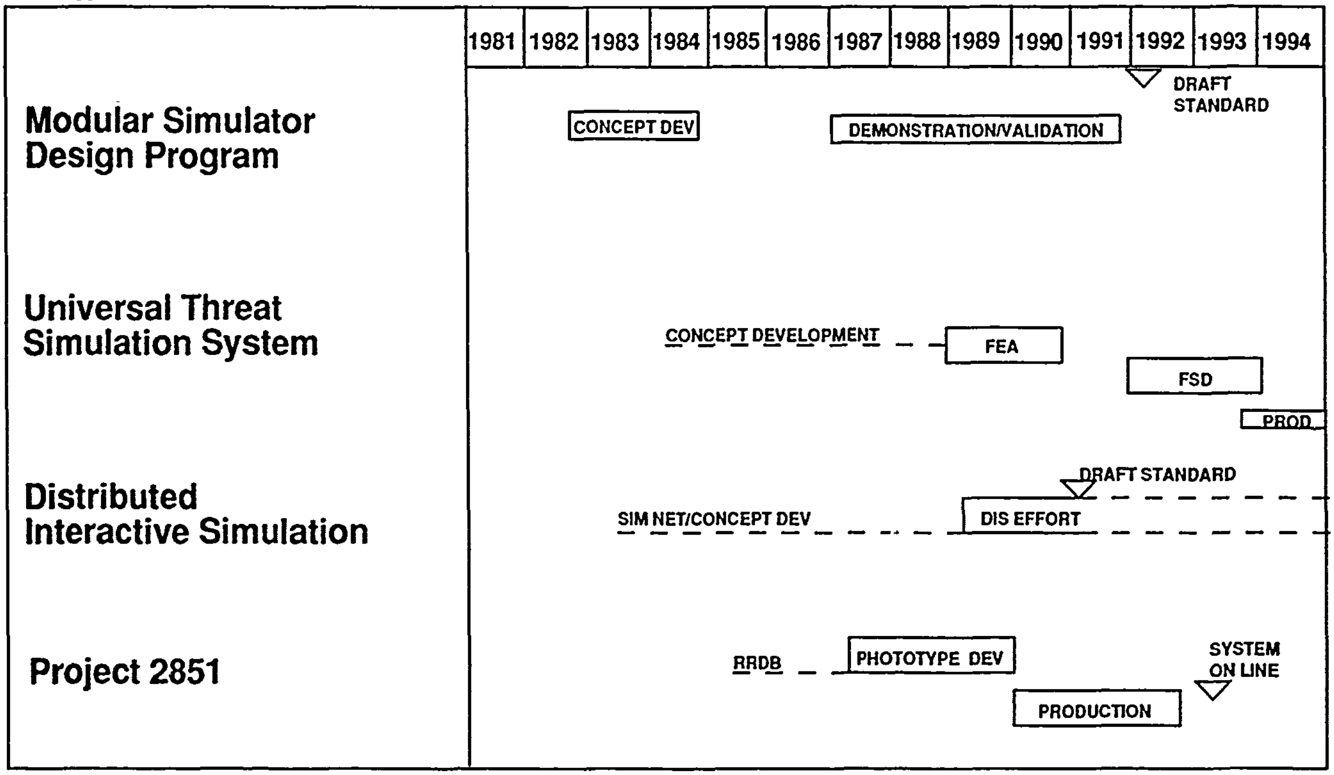


FIGURE 7
Standardization Program Development Schedule

Another problem among the standardization programs is redundancy in solving technical problems which are common to the programs. An example of this is the determination of data units for interfaces. One of the goals of each standardization program is to provide a standard interface for data to the external world. Each program has developed this interface and each program has defined a standard set of data units for the interface. The development and definition of data representation and data units was a significant and highly debated issue in each program. However, in most cases the units among the programs don't match. This will cause data conversions to be performed when more than one of the standards is invoked for a single program. These data conversions are a needless waste of valuable computational resources. As part of a study, a one to one correlation was conducted between the MSDP interfaces and the DIS interfaces. It was found that almost all data units required some conversion.

In some cases there is a specific reason for the data units to be defined as they are for a certain program, such as packet size, data latency, etc., but the determination of units on several programs were based on the standard developer's initial input.

The Environment

The lack of correlation of data units among the programs is of minor importance when compared to how the technical aspects of the programs will interact when their associated standards are used together. What is needed is a global view of the programs to determine how the standards produced by the programs may eventually interact.

If a comparison is made between the real world and the simulated world, a simple correlation can be derived as shown in Figure 8 for a single simulation device. The simulation can be abstracted into three major components in the simulated world, 1) The control function or Instructor Operator Station (IOS) in most simulators can be compared in a simplistic sense to God who has the power to control the environment and anything in the environment including the ownship, 2) The environment function which is defined as anything external to the ownship, and 3) The ownship itself which contains all ownship parameters specific to the aircraft being simulated. One concept behind cost reduction is to maximize reusability by identifying those areas of the simulation which are generic (reusable) and those areas which are application specific (non-reusable). An

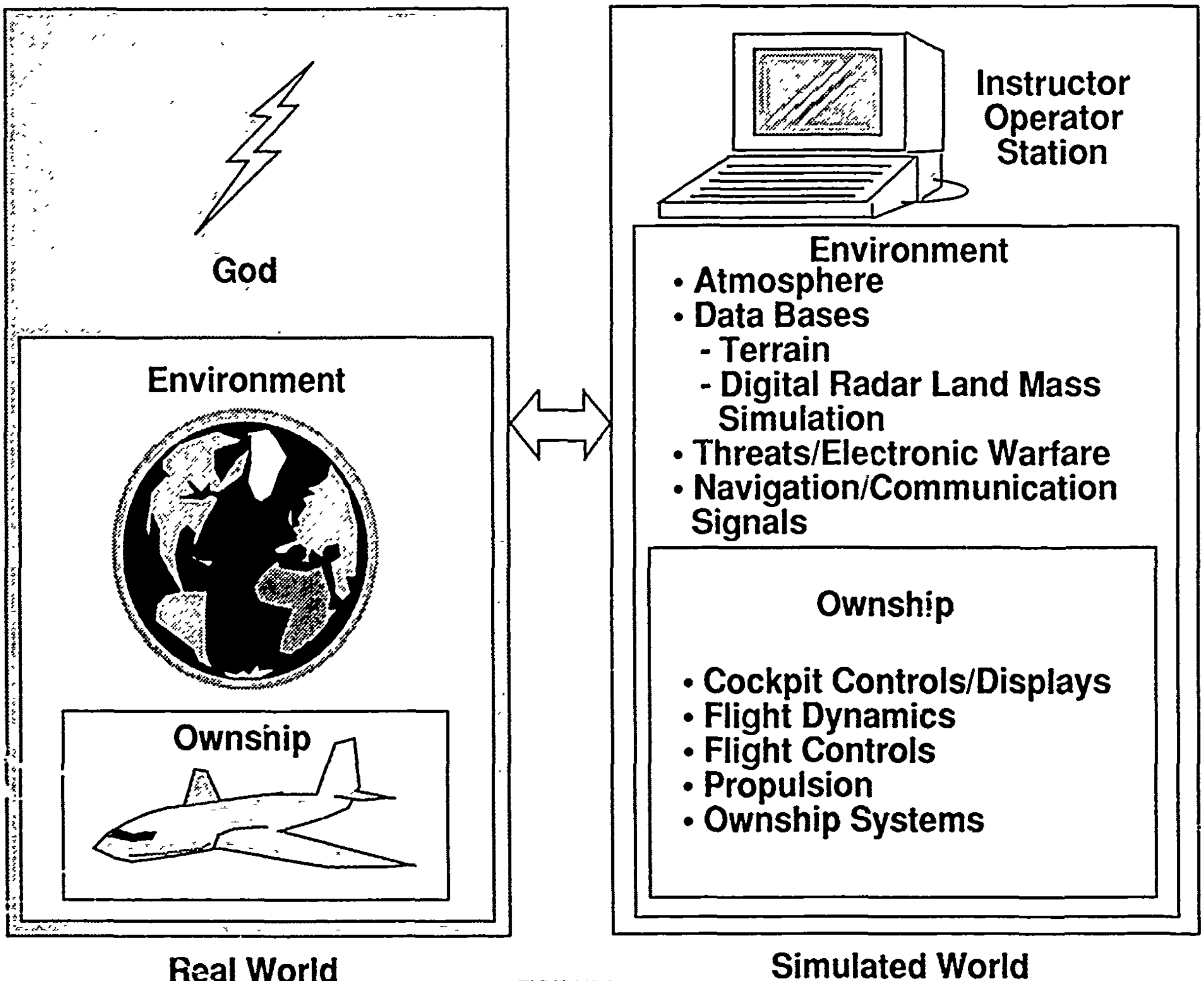


FIGURE 8
Real World to Simulated World Comparisons

environment simulation could be considered reusable for any ownship. The environment simulation can remain constant, whereas the ownship simulation is application specific and must be replaced for a particular application. If the interfaces between the three major components are clearly defined and included as part of the standards, this partitioning would allow a great deal of flexibility among the standards programs.

To align to the common environment concept a Modular Simulator could be partitioned as shown in Figure 9. In this configuration an Environment module has been added to the current Modular Simulator standard allocation. The functions internal to the modules have been reallocated such that the remaining modules with the exception of the IOS module contain only the ownship unique functions. The entire environment would reside in the environment module. This environment would include the tactical realm of threats and electronic warfare as well as the natural atmospheric environment. In this partitioning the Modular Simulator would get the environmental simulation from an internal environment when in a stand-alone simulation mode and from the DIS when in a networked, or multiple simulator, simulation mode. Also when in the stand-alone mode the UTSS would be a part of the Environment module. In the DIS or networked mode the Environment module would serve two primary functions. The first function would be a translator between DIS messages (PDUs) and the appropriate Modular Simulator internal message structure. The second function would be to provide those environment functions not available on the DIS network. Currently the DIS only provides entity states and a limited electronic warfare capability. As DIS grows and these functions are added then the Modular Simulator Environment module can be modified.

An additional benefit to this partitioning is that changes in the DIS and UTSS standards have been isolated to a single module in the Modular Simulator. The interface from the Environment module to the MSDP global bus can be completely defined and should not be affected by changes in the other standards.

To expand the common environment concept further, an Environment module could be

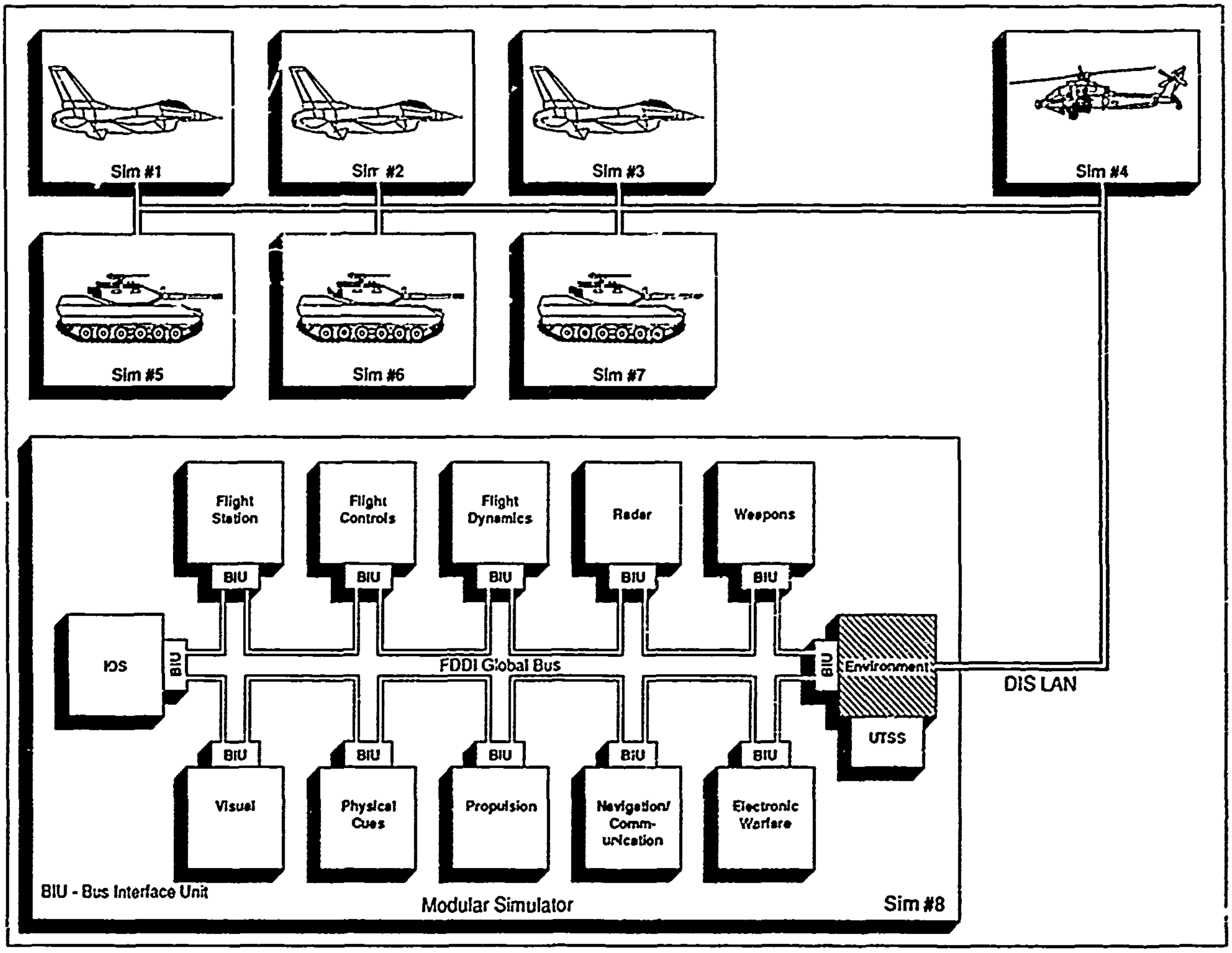


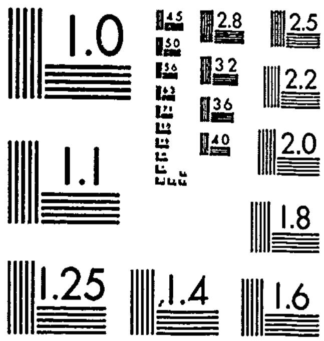
FIGURE 9
Modular Simulator Environment Concept

added to the DIS network as shown in Figure 10. In order to demonstrate this, assume that the DIS would provide the complete system including ownships and the environment. In a networked mode each ownship simulator would view other ownships as part of the environment and would receive common environmental information from the DIS Environment module. The UTSS data base could be added to the DIS Environment module if a threat environment was required in addition to the ownships connected to the DIS network. Furthermore, standard data bases from Project 2851 could be stored in the DIS Environment module to be used in the DIS exercises. With respect to the use of Project 2851 two methods could be employed. The data bases could be shared and accessed during run time for dynamic changes to the data base or the data bases could be copied with copies residing at each ownship and dynamic changes transmitted via the DIS network. The latter would probably be the most feasible solution. However, both methods would require further study to derive a sound technical solution.

This concept does not take into account two issues; the incorporation of existing simulators into the DIS network, and the relative fidelity of training devices that are interacting in the DIS network.

Many existing simulators are not partitioned in the same manner as a Modular Simulator and do not have a discrete Environment module to act as the system interface. This will cause a significant rework to the existing simulator. However, in all fairness many existing simulators cannot easily accommodate an interface to UTSS or DIS without a considerable amount of rework. Therefore, the common environment concept would not impose any additional rework to an existing simulator over what is now required to interface to these systems.

The relative fidelity issue, simulators of high and low fidelity operating in the same training exercise and communicating with a common environment, is also not impacted to any greater degree by this design concept. The relative fidelity issue will probably not be completely solved by any standards program. The simple truth is that mixing devices of varying fidelity will be a very difficult if not impossible task without some rework of the devices to allow multiple fidelities to interact.



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

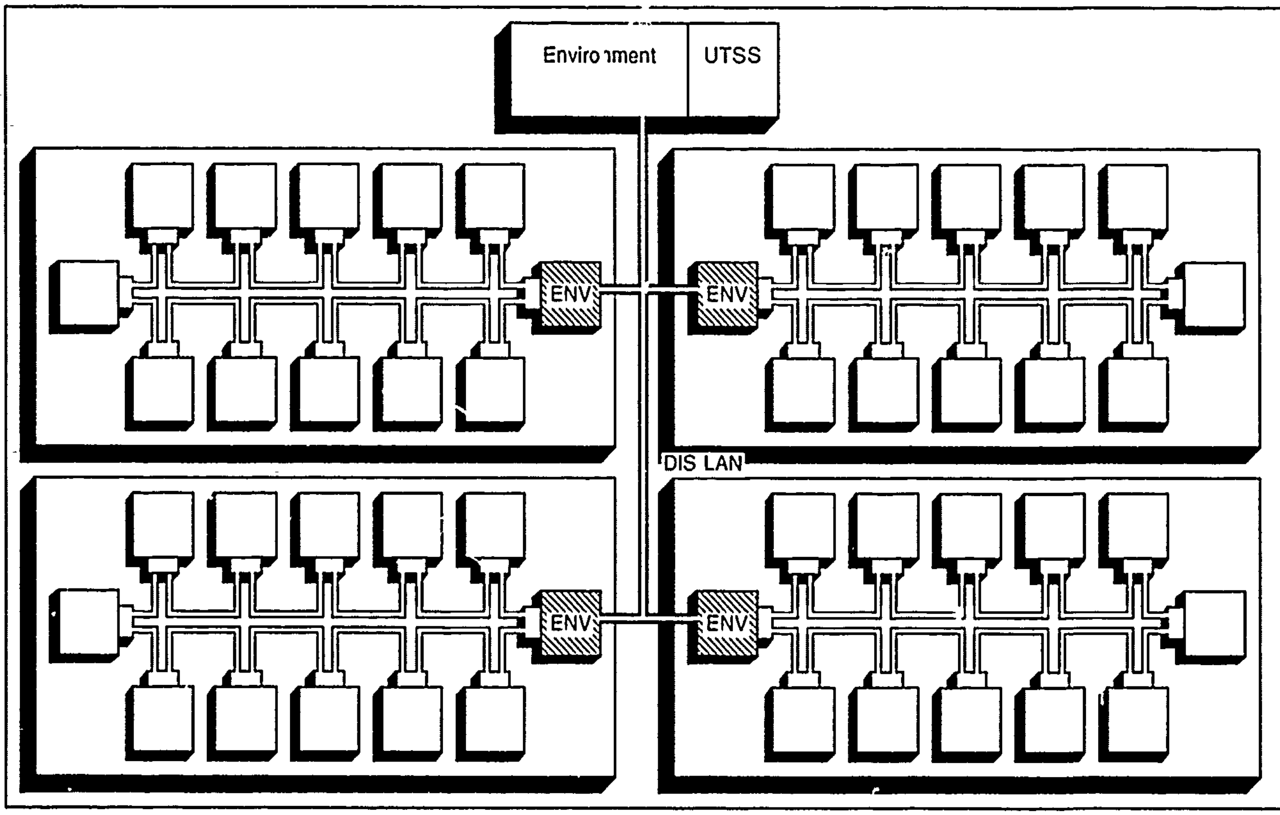


FIGURE 10
Distributed Interactive Simulation Environment Concept

What the common environment concept does provide is the isolation of change in the interface between all standards efforts. All changes in the DIS or UTSS standards have been isolated to the Environment module in the Modular Simulator standard. Changes to the UTSS or Project 2851 standards have likewise been isolated to the common environment of the DIS standard.

RECOMMENDATIONS

Based on the data presented in this paper several recommendations can be made for changes or considerations for change in the various standards programs. This is not intended to be an exhaustive list but a starting point for further analysis of the interaction and interface among the various standards programs.

Modular Simulator Design Program

To better align with the team training/multiple networked simulator environment supported by the DIS standard, it is suggested that the existing Modular Simulator functions be repartitioned and new functions be added as required to allow for an Environment module. This would allow for a single point of connection to a DIS environment and a more flexible design with respect to changes in both the emerging DIS and UTSS standards. The remaining modules should be repartitioned to reflect only ownship functions with the exception of the IOS module which would retain its current functionality.

Universal Threat Simulator System

The UTSS is still early in its development cycle. This allows UTSS to make use of the existing designs and lessons learned from MSDP, Project 2851 and DIS to develop an interface that is compatible with the standards produced by these programs and still meets the unique requirements of UTSS. Since UTSS will be dealing with the same type of entities already defined in the DIS standard, it is suggested that UTSS attempt to conform to the DIS PDU structure for entity descriptions and units wherever possible to allow for an interface which does not require a significant amount of data transformation. This would also allow UTSS to directly connect a threat environment to the DIS network in a seamless manner. For the sake of consistency among standards, UTSS should also consider using the Project 2851 system methodology for the threat models and database generation.

Distributed Interactive Simulation

Although the DIS standard has defined the basic PDUs for the interoperability of training devices there is still a great deal of work to be accomplished before DIS

is a complete standard. It is recommended that DIS consider the environment module concept in its design along with the ability to interface directly with the UTSS and Project 2851 standards.

Project 2851

There are no recommended changes to the Project 2851 standard. This standard does not have a global interface impact to the other standardization efforts. The interfaces to this standard have been defined and the process of using the Project 2851 system for storage and validation of existing database information and generation of databases for future trainers is established. The Project 2851 standard should be invoked as soon as possible to take advantage of the system.

Common I/SWG

It is suggested that as the draft standards become available, a central organization should review the standards for mismatches. If possible, this organization should attempt to resolve disconnects including redundant specifications and areas where further specification is required. Such work promises to provide a well defined interface among the standards. This task could be the effort of the joint or common I/SWG. What is needed is a technically competent group that has a good technical working knowledge of each program and can take an open minded, objective look at each program to ensure that the programs mesh together.

This I/SWG could be composed of individuals from the standardization programs, selected members of industry and the government. The members should be "active" participants in the existing I/SWGs if possible to provide an interface between the common I/SWG and the program I/SWGs. If possible this I/SWG should be funded so that the members might consider their efforts a part of their regular jobs and not an extracurricular activity.

CONCLUSION

The government and industry efforts to standardize certain aspects of simulation and training technology will eventually lead to improved training and simulation tools. The end result would be improved operational readiness with less technical risk and lower training costs. In most respects the four standardization programs identified in this paper interface quite well at their current state. However, these programs should be periodically reviewed to ensure that they continue to interface with each other to provide viable standards. If conflicting standards are produced it is possible to unintentionally increase the technical risks and costs associated with future training systems.

The concepts of a common I/SWG and development of a common environment should be seriously considered in the creation of these standards, particularly for the MSDP and DIS programs. These programs will derive the greatest benefit from a common environment, particularly the DIS standard, which deals with combined forces operations. By incorporating these emerging technologies into the standards produced today it will be possible to sustain and continue to improve training technologies within the decreasing defense budgets of the future.

ABOUT THE AUTHOR

Mr. Gary M. Kamsickas is a Software/Systems Engineer with the Simulation and Training Systems organization of Boeing Defense and Space Group in Huntsville, Alabama. He has been responsible for software design, code, test and integration on several Boeing simulator projects, including the Ada Simulator Validation Program (ASVP) and the Modular Simulator Design Program (MSDP). He is currently the Principal Engineer for the MSDP and involved in the design validation of the modular simulator concept. Mr. Kamsickas holds a Bachelor of Science degree in Electrical Engineering from Michigan Technological University, Houghton, Michigan.

A MODEL FOR COMPUTER-BASED TRAINING QUALITY ASSURANCE

Michael Quattrociocchi
Linda Donegan
GE Aerospace
Syracuse, New York

ABSTRACT

This paper details a Quality Assurance (QA) plan for interactive Computer-Based Training (CBT) to ensure quality is an element inherent during all phases of production. The cornerstone of the plan is comprised of quality assurance measures incorporated into all aspects of the CBT lesson production process extending from the very earliest stages of development to the point of final delivery. Specific stages of production have been identified as effective times/places for auditing the actual process and are referred to as "QA Checkpoints." These checkpoints provide an opportunity to verify the product quality while checking for adherence to process. Items examined include material tracking and control, documentation, and courseware availability for review by appropriate contractor or client parties. This model plan can be a valuable instrument in producing an optimum product while controlling costs, and offers a foundation for varied applications across the CBT industry.

INTRODUCTION

Implementation of a cost-effective Quality Assurance and Control Plan is a growing concern to all program managers. The consequences of an ill-devised plan can range from significant cost growth to inability to achieve a sale for CBT material.

It is not the intent of this paper to discuss all aspects of CBT development. Therefore, prior to a detailed discussion of the QA model, certain assumptions must be stated. First, fundamental to any successful CBT program is a well-defined set of design standards to include: screen design/formats, use of color, use of text, templates to implement instructional strategies, scripting structures, etc. Secondly, to implement a successful QA model assumes that a well-trained team has been established. Team members consist of subject matter experts, programmers, graphics developers, instructional system designers, authors, editors and video producers. Depending on the scope and talents of the team, project members may perform more than one responsibility. Further, it is the experience of the authors that a highly productive team must be cultivated and nurtured. The QA model presented here was developed to support various moderately complex interactive CBT programs. These programs required the use of full motion video, still video, text, audio and graphics for lesson presentation. The primary goals in the design of the QA model were to:

- Provide an efficient, easy-to-follow CBT development process.
- Control implementation of changes to courseware; continuous uncontrolled change results in added cost and reduced quality.
- Provide customer confidence in the product quality.

Figure 1 presents an overview of the CBT lesson development process; a description of process components follows.

Course Design

The course design defines which lessons will be presented to the student via CBT. Objectives, content capsules, and detailed outlines are developed to provide the foundation upon which the subject matter for each lesson will be based.

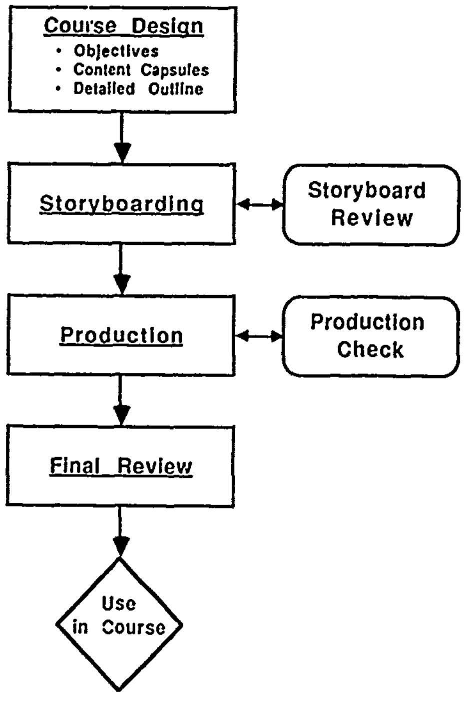


Figure 1. CBT Lesson Development Process

Storyboarding

The core team members who develop the lessons are referred to as storyboard authors. The authors write the instructional text, and specify the content and placement of all graphics, video and audio components required for the lesson. The output of this process is the storyboard: a complete specification sheet for the lesson.

Storyboard Review. The storyboard is passed to a team of reviewers who ensure the quality and integrity of its components. The team consists of specialists in the various areas of CBT lesson development.

Production

The term "production" as it relates to storyboards refers to the programming, graphics, and video development work required to meet the specifications contained in the storyboard.

Production Check. Before the computer presentation of the lesson is forwarded to the final review team, a complete production check is done. This review is to ensure that all production elements have been properly implemented.

Final Review

The final check is a computer presentation review of the lesson by contractor personnel. It is the last review prior to course conduct.

PROCESS TOOLS AND PARAMETERS

Quality Assurance Checkpoints

To meet our goals in the design of a QA model, specific stages of development have been identified as effective times or places for auditing the process. These are referred to as QA checkpoints. See Figure 2.

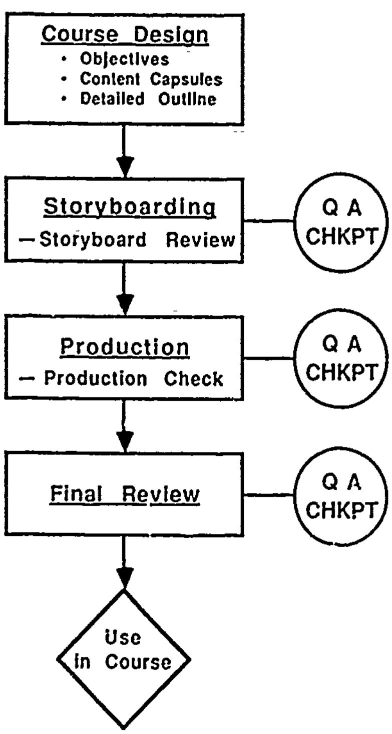


Figure 2. Quality Assurance Checkpoints

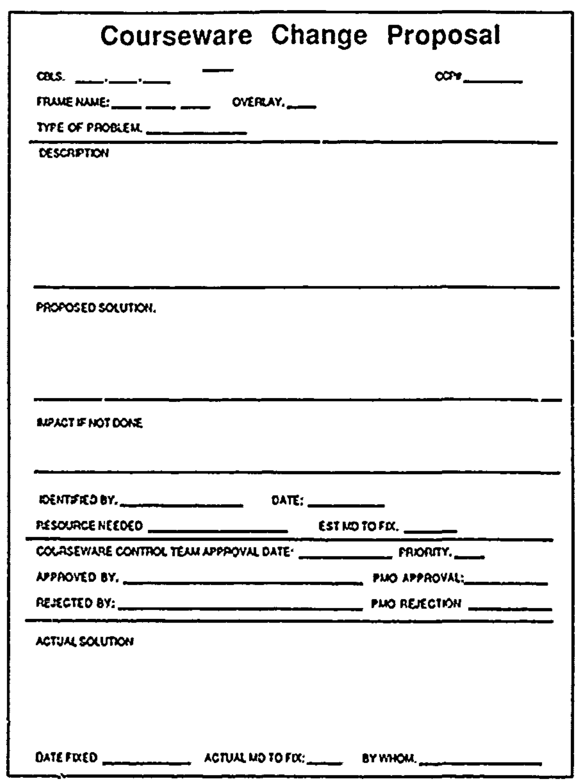
All QA checkpoints provide the opportunity for an audit of the process itself to ensure that:

- production is being carried out as planned
- supporting documentation is complete and up-to-date
- material tracking records are in order
- materials and documentation are available for review
- materials meet established design and visual standards

If a QA checkpoint audit were to reveal a deviation from the prescribed process, this would indicate a need to closely assess the situation and consider possible flaw(s) in the process. The QA functions may interrupt or stop production activities if, in the judgment of the QA coordinator, such action is warranted by a violation of the process. Continual scrutiny of the process is necessary to ensure its effectiveness; QA checkpoints provide the opportunity for such scrutiny. Audits at the designated QA checkpoints may or may not always take place, particularly by external/client parties. However, the opportunity is availed at each QA checkpoint.

Corrective Actions & Documentation

All corrective action items, regardless of where in the production process they occur, are addressed using a Courseware Change Proposal (CCP) form, Figure 3. A proposed change is submitted on a CCP form by a problem identifier, who may represent any contractor team or



Courseware Change Proposal

CCP#

FRAME NAME: OVERLAY.

TYPE OF PROBLEM

DESCRIPTION

PROPOSED SOLUTION.

IMPACT IF NOT DONE

IDENTIFIED BY. DATE:

RESOURCE NEEDED EST MD TO FIX.

COURSEWARE CONTROL TEAM APPROVAL DATE: PRIORITY.

APPROVED BY. PMO APPROVAL:

REJECTED BY: PMO REJECTION

ACTUAL SOLUTION

DATE FIXED ACTUAL MD TO FIX: BY WHOM.

Figure 3. Courseware Change Proposal Form

REFERENCES

[1] Akeley, K. and T. Jermoluk, "High-Performance Polygon Rendering," SIGGRAPH '88 Proceedings, Computer Graphics, V. 22, No. 4, August 1988, pp. 239-246.

[2] Barkans, A. C., "High Speed High Quality Antialiased Vector Generation," SIGGRAPH '90 Proceedings, Computer Graphics, V. 24, No. 4, August 1990, pp. 319-326.

[3] Blinn, J. F., "Return of the Jaggy," IEEE Computer Graphics and Applications, V. 9, No. 2, March 1989, pp. 82-89.

[4] Bracewell, R. N., The Fourier Transform and its Applications, McGraw-Hill Book Company, New York, 1978.

[5] Brassel, K. E. and R. Fegas, "An Algorithm for Shading of Regions on Vector Display Devices," SIGGRAPH '79 Proceedings, Computer Graphics, V. 13, No. 2, August 1979, pp. 126-133.

[6] Bresenham, J. E., "Algorithm for Computer Control of a Digital Plotter," IBM Systems Journal, V. 4, No. 1, 1965, pp. 25-30.

[7] Carpenter, L. "The A-buffer, an Antialiased Hidden Surface Method," SIGGRAPH '84 Proceedings, Computer Graphics, V. 18, No. 3, July 1984, pp. 103-108.

[8] Cook, R L., "Stochastic Sampling in Computer Graphics," ACM Trans. cn Graphics, V. 5, No. 1, January 1986, pp. 51-72.

[9] Crow, F. C., "A Comparison of Antialiasing Techniques," IEEE Computer Graphics and Applications, V. 1, No. 1, January 1981, pp. 40-48.

[10] Crow, F C , "Antialiasing" in Computer Graphics Techniques - Theory and Practice, D F Rodgers and R A. Earnshaw, Eds Springer-Verlag, New York, 1990.

[11] Gish, W., and A. Tanner, "Antialiased Scan Conversion of Lines and Polygons," submitted for publication.

[12] Gupta, S. and R F Sproull, "Filtering Edges for Grey-Scale Displays," SIGGRAPH '81 Proceedings, Computer Graphics, V. 15, No. 3, August 1981, pp. 1 7.

[13] Haeberli, P. and K. Akeley, "The Accumulation Buffer: Hardware Support for High-Quality Rendering," SIGGRAPH '90 Proceedings, Computer Graphics, V. 24, No. 4, August 1990, pp. 309-318.

[14] Kajiya, J. T., "The Rendering Equation," SIGGRAPH '86 Proceedings, Computer Graphics, V. 20, No. 4, August 1986, pp. 143-150.

[15] Krantz, J. H., and L. D. Silverstein, "The Grey-Scale Asymptote for Anti-Aliasing Graphics Images on Color Matrix Displays," Soc. for Information Display 89 Digest, pp. 139-141.

[16] Leler, W. J., "Human Vision, Anti-Aliasing, and the Cheap 4000 Line Display," SIGGRAPH '80 Proceedings, Computer Graphics, V. 14, No. 3, July 1980, pp. 308-313.

[17] Mitchell, D. P. and A. N. Netravali, "Reconstruction Filters in Computer Graphics," SIGGRAPH '88 Proceedings, Computer Graphics, V. 22, No. 4, August 1988, pp. 221-227.

[18] Naiman, A. C. and W. Makous, "Information Transmission for Greyscale Edges," Soc. for Information Display 91 Digest, pp. 109-112.

[19] Porter, T. and T. Duff, "Compositing Digital Images," SIGGRAPH '84 Proceedings, Computer Graphics, V. 18, No. 3, July 1984, pp. 253-259.

[20] Szabo, N. S., "Digital Image Anomalies: Static and Dynamic," in Computer Image Generation, B. J. Schacter, Ed., John Wiley & Sons, New York, 1983.

[21] Yan, J. K., "Advances in Computer-Generated Imagery for Flight Simulation," IEEE Computer Graphics and Applications, V. 5, No. 8, August 1985, pp. 37-51.

ACKNOWLEDGEMENTS

The authors wish to thank Dr. Frank Crow of Apple Computer for his permission to use the images in Figure 4. We would also like to acknowledge the continuing support of Lorin Bice, President, and everyone else at Terabit for making this work possible.

ABOUT THE AUTHORS

Walter Gish is Executive Scientist at Terabit Computer Engineering. He is responsible for the development of advanced visual systems. Mr. Gish holds BS and MS degrees in Physics and Applied Mathematics from the California Institute of Technology.

Allen Tanner is the founder and Chief Executive Officer of Terabit Computer Enginerring. He has designed and implemented a wide variety of computer image generation and video conversion systems. Mr. Tanner received his BS in Computer Science from the University of Utah.

Storyboarding

Core teams (A), comprised of an instructional designer and one or more subject matter experts, use lesson specifications to write instructional text and specify placement of all graphics, audio and video components for a lesson. The product of this process is a storyboard. Upon completion, the core team submits the first draft of a storyboard to a team of reviewers who ensure the quality and integrity of its components.

The review team (B) consists of specialists who review the storyboard guided by criteria in checklists for the following areas:

- instructional design
- technical content
- graphics and video production requirements
- text composition, spelling, and grammar
- CBI production feasibility

Figures 5 and 5A are sample review checklists. This team then either approves the storyboard (C), makes minor changes required for approval (C), or returns the storyboard to the core team for corrective action (D). The storyboard is passed between the core team and review team as many times as required to bring it to acceptable baseline standards.

Once the storyboard is approved by the review team, it is put under control of the data librarian (E) and a baseline is achieved (F). As a lesson moves through the different phases of the process, its status is tracked by a data librarian for control purposes. Sign-out sheets are used to track the physical location of the lesson at any given time prior to delivery, and are monitored periodically by the QA function. Review team sign-off sheet for the storyboard, and the storyboard itself are available for the first QA checkpoint (G).

GRAPHICS REVIEW CHECKLIST LESSON:

	YES	NO
1. Is there a request form for each primary graphic? Is a drawing or sketch included or referenced? Description, colors, and all other necessary information on sheet?		
2. Is the graphic to be used for interactions? Indication on form? Are necessary supporting graphics called out for the interactions indicated? Touchzone(s) identified (specific area)?		
3. Is a primary graphic request form referenced for each overlaying secondary graphic?		
4. Are the notes clear enough to complete the graphic and the interaction?		
5. Are references to existing graphics included?		
6. Will the graphic fit into the visual area for the screen type (oriented correctly)?		
7. Is the draw time estimated to be 10 seconds or less?		
8. If narration is used to support the graphic, Is is really necessary? Is it cost effective?		

Figure 5. Sample Checklist

Production

Next, the storyboard goes to production. Production includes the development of graphics, audio, video, and coding required to meet storyboard specifications. A team of computer graphics artists draws the images and diagrams to accompany the text presented to the student. Likewise, the video production team, under the supervision of a video coordinator, shoots the scenes and motion sequences that will demonstrate the actual performance of a given task.

All corrective actions are handled via the CCP process (H). Proposed changes and actual changes are documented and the storyboard is returned to the data librarian. The final step in production is a post production check (I). This review is to ensure that material was produced as specified in the storyboard.

This review is conducted by a team member who is familiar with production techniques and acceptable visual standards. This person should not be directly involved in the actual development and integration of the video, graphics, audio and programming elements. Any corrective actions which emerge as a result of this production check, are addressed per the CCP process (H). Problems are discussed, documented, and corrected. Once again, the lesson is placed under control of the data librarian. Production sign-off of the storyboard and CCPs are supporting documentation for the second QA checkpoint (J).

INSTRUCTIONAL DESIGN CHECKLIST LESSON:

	YES	NO
1. Has the lesson objective been met (to the extent possible using CBI)?		
2. Are the interactions meaningful and appropriate?		
3. Is the cueing consistent and effective? Appropriate level Consistent Effective Appropriate amount		
4. Are criterial components of instructional messages emphasized? Noncriterial elements de-emphasized? Does the text match the visual idea?		
5. Does the organization of the lesson support client's cognitive style?		
6. Are the video, audio, text, and graphics used appropriately? Do they support the learning objectives?		
7. Are video, audio, text, and graphics combinations used appropriately to emphasize criterial components of the message?		
8. Are questions written clearly and at the appropriate level? Relate to learning objectives? Follow format guidelines? At the appropriate knowledge and skill levels? Stem - unambiguous? Distractors - plausible, unambiguous, effective?		

Figure 5A. Sample Checklist

Final Review

This review is performed in two stages. The first stage is the instructional design/subject matter expert (ID/SME) review (K). It is conducted as a joint review performed at a shared terminal. The ID/SME team reviews the entire lesson, frame-by-frame, in an effort to uncover presentation problems. These problems may be technical inaccuracies, coding errors, misrepresentations, or instructional approaches that are for some reason not effective. Problems or issues are documented and corrected following the established CCP corrective action process (H). The necessary corrections are made and the lesson is forwarded to the second stage of the final review (L). The lesson, at this point, exists exactly as intended for the student.

This review is performed by two QA team members. One member has a technical background similar to that of the actual student. The second member is familiar with the operation of the CBT delivery system and able to proceed through the lesson.

The final QA team employs techniques to determine if the lesson "works." These two team members proceed by viewing the lesson exactly as the student would. They also intentionally answer questions and respond to prompts incorrectly to determine if the system reacts as intended. They conduct an overall inspection of the lessons, checking for instructional flow, grammar, and proper integration of graphics, video, audio and text. Corrective actions are handled via the CCP process (H) and changes are implemented as necessary.

Completion of the final QA review is the third QA checkpoint (M). Completed, up-to-date lessons can be monitored for compliance to established procedures at this point. All records and data maintained for effective production operations should be available for review, including storyboards, various sign-off sheets, checklists and documentation of corrective action issues. Copies of any individual records will be furnished to the client upon request. It is the responsibility of the final QA team to assure that records are up-to-date, complete and reliable. When all changes are incorporated and final QA team approves the lesson, it is ready for client review. The lesson is, again, placed under control of the data librarian (N).

External (Client) Review

Upon completion of the final QA review, the lessons are passed to external baseline (O). Lessons are ready for client review and, ultimately, course use. Client review is the fourth and final QA checkpoint (P) that provides an opportunity to audit the process. All accompanying documentation should be complete, accessible, and should support the lesson exactly as it was produced.

Any changes to a lesson beyond external baseline are subject to a CCP process (Q) administered by program management. Documentation of all CCPs submitted beyond the external baseline is made in a master control file (R). Use of a Master Control File (MCF) provides an efficient method to review corrective action proposals submitted by the client.

Lessons are maintained under control of the data librarian until delivery to the customer.

LESSONS LEARNED

- Process seems very cumbersome at times, but perseverance/adherence to it proves beneficial; the process is very thorough.
- Standards must be well-defined from the start. This facilitates/lends consistency to materials and eliminates subjective guesswork at various review stages. Also saves time and unnecessary debate in CCP meetings.
- Take no shortcuts in the paperwork trail; numerous times, due to the volume of material being produced and decisions being made, only the documentation told the story. Team members often couldn't remember why something was handled as it was, or where a particular storyboard or lesson was at a given time, and for what reason.
- Always assess the "domino effect" of a proposed change and the value and necessity of that change vs. the effort involved in making the change.
- QA is something to be addressed from start to finish; it's not a final stage in production. Team members need to be in this mindset from the very start.
- Strict adherence to standards and to the process from the first stages will save a lot of time at the final stages. Problems should be minimal by the time the lesson reaches final review. The time saved is actually money saved, and the quality is better if it's built in and maintained from the start.

SUMMARY

There are many variables when managing different computer-based training programs. Perhaps the two most important aspects are the unique requirements of different programs and the unique personalities of the CBT development team. It is the program manager's responsibility to orchestrate a successful program.

Use of the QA program presented here is a foundation to building a successful program. It should be molded to fit specific program needs, and the capabilities and personalities of team members.

Use of this QA plan has yielded favorable results. Development time for comparable CBT material was reduced by an average of 20%. In addition, the percentage of errors identified during the client review prior to course conduct was less than 1%.

REFERENCES AND SUGGESTED READING

1. MIL-Q-9858A-Quality Program Requirements

2. MIL-STD-1520B-Corrective Action and Disposition System for Nonconforming Material

3. MIL-STD-153A-Supplier Quality Assurance Program Requirements

ABOUT THE AUTHORS

MICHAEL QUATTROCIOCCHI is the Manager of Technical Training for the Radar Systems Department of GE Aerospace. Mr. Quattrociocchi has worked in domestic and international government contract training for more than eight years. His experience spans all areas of instructional systems design with particular emphasis on interactive computer-based training and simulation equipment. He has a Master of Science degree in Instructional Design/ Development and Evaluation from Syracuse University.

LINDA DONEGAN is currently performing the editing and quality assurance functions for the production of training publications to be used during conduct of several customer courses on GE-manufactured radar systems. Her duties include defining and enforcing quality standards for courseware development, coordination of production efforts, and critique of stand-up instruction presentations. Prior to this assignment, she was responsible for co-authoring, implementing, and coordinating the quality assurance process applied to GE's production of interactive computer-based training lessons for Saudi Arabian students. She has a Bachelor of Arts degree in Communication Arts from the State University of New York at Plattsburgh.

U. S. ARMY MATERIEL COMMAND'S INTELLIGENT TUTORING SYSTEM TECHNOLOGY BASE PLAN

Willard M. Holmes
U.S. Army Missile Command
Research, Development, and Engineering Center
Redstone Arsenal, AL 35898

Admiral Piper
Office of Project Manager for Training Devices
Orlando, FL 32826

ABSTRACT

The Army Technology Base Master Plan identifies the emerging field of artificial intelligence as a technology with high potential to meet the Army's changing needs for fixing, manning, and arming the forces into the next century. In late 1990, the Army Materiel Command's Deputy Chief of Staff for Technology Planning and Management directed a comprehensive master plan for artificial intelligence be developed. This plan serves as a framework from which the Army Materiel Command can manage and execute AI technology development into the 21st Century. The AI Master Plan identifies 13 technology areas considered most relevant to the Army needs. This paper addresses one technology area in the AI Master Plan, Intelligent Tutoring Systems. This is a plan within the master plan and provides a vision and specific direction for closing the gap between today's reality and tomorrow's expectation for Army training systems.

INTRODUCTION

In April 1989, the Army published the Army Technology Base Master Plan. The Plan provided guidance to the Army Laboratories and Research Centers to focus the technology base on the most critical war fighting needs. The plan directive is to support force modernization programs while preserving and enhancing our technological superiority over potential adversaries. The Technology Base Master Plan recognized the technology base as an essential corporate investment in the Army future. It also contains the Technology Base Investment Strategy for realizing the leadership's vision of the future Army support for Commander-in-Chief warfighting needs.[1] By identifying a particular area as a key emerging technology, the Army signals its intention to (1) provide sufficient funding for progress on a broad front, (2) stabilize this funding so that laboratory activities can be planned properly, (3) ensure that the technical staff has developed concrete plans, and (4) provide a mechanism by which management can review important areas across organizational boundaries.

Artificial Intelligence (AI) has been identified and designated in the Technology Base Master Plan as one of the 13 key emerging technologies that have been recognized as having a greater impact than other technologies on future warfighting capabilities. The combination of AI being recognized as a critical emerging technology and the specific need for a concrete plan is the motivation for the development of the Army Materiel Command's (AMC's) AI Master Plan.

Given these stated conditions, the objective of this paper is first to inform other government agencies and industry that the Army's AMC is taking decisive steps to organize and focus on AI based technology based activities. A second objective is to review a conceptual framework for conducting research and development on Intelligent Tutoring Systems (ITS). A third and final objective is to provide a brief vision of intent for advancing and applying ITS technology development to meet Army training needs into the 21st Century.

ARMY MATERIEL COMMAND'S AI TECHNOLOGY BASE MASTER PLAN

Objectives

The AMC AI Master Plan addresses the contribution that AI technologies can make to the AMC and Army systems. It emphasizes uniformly the AI technology development, transition, and application phases. It encompasses the research, development, and application activities, as well as, the application of AI to manufacturing, testing, and logistics. The broad objectives of the master plan serves the following purposes:

a. Construct a framework for AI research and development in the organization,

b. Serve as inputs to the AMC Technology Base planning,

c. Catalogue significant current and future AI and AI related projects,

d. Identify AI resources availability and needs,

e. Coordinate all AI projects and activities internally within and externally among various organizations,

f. Demonstrate paths to acquiring skills, services, products, and facilities,

g. Evaluate offers of help from outside agencies,

h. Show ways of applying AI in devices, systems, and services,

i. Demarcate the role of AI by AMC missions, objectives, and goals,

j. Identify funding needs to realize full gain from AI,

k. Identify synergy of AI with other technologies and projects,

l. List benefits and risks of AI systems.

Emerging AI Technologies

In 1956, a group of scientists assembled at Dartmouth College coined the term "Artificial Intelligence" to describe a vaguely defined but emerging technology. The collective initiatives of this group of pioneering scientists resulted in what today is identified as Artificial Intelligence or AI. During the almost four ensuing decades some elements of artificial intelligence have made the transition from the often mysterious topic reserved for laboratory and university research to commercial applications. While most of the early prediction on the achievements of AI fall short of expectations, some areas of the emerging technology have achieved widespread recognition. Key application areas of AI technology include: expert systems, computer vision, natural language processing, speech interfaces, problem solving and planning. Expert systems was the first sub-component of AI technology to achieve a moderate amount of commercial success. The successful development and application of expert systems technology is detailed in The Rise of the Expert Company.[2]

During the last decade DARPA's Pilot Associate and Battle Managements Programs[3] are examples of programs that are focal points of intense development and application of AI technologies. More than 40 Army centers, laboratories, and agencies have indicated either an interest or have AI related activities in ongoing program efforts.

In early 1989, AMC leadership initiated the effort to develop the AI Master Plan. Major subordinate commands were asked to provide inputs to the plan. Technology areas most relevant to the Army needs were defined during ensuing workshops, working sessions, high level straw man plans development, product and specific technical area reviews. The thirteen AI technology areas determined to be most relevant to the Army's needs into the 21st Century are indicated in Table 1. As shown, some technology areas include more than one technology. As expected, the boundaries between the technology areas overlap considerably.

ARTIFICIAL INTELLIGENCE TECHNOLOGIES MOST RELEVANT TO ARMY NEEDS
1. EXPERT SYSTEMS KNOWLEDGE LEVEL REASONING, MACHINE LEARNING
2. NATURAL LANGUAGE AND SPEECH RECOGNITION AND INTELLIGENT INTERFACES
3. INTELLIGENCE VISION AND IMAGE UNDERSTANDING
4. INTELLIGENT DATABASE AND COMMUNICATIONS
5. INTELLIGENT SENSOR AND DATA FUSION
6. AUTONOMOUS SYSTEMS AND INTELLIGENT CONTROL
7. INTELLIGENT PLANNING
8. INTELLIGENCE SIMULATION
9. AUTOMATIC PROGRAMMING
10. INTELLIGENT MANUFACTURING AND CONCURRENT ENGINEERING
11. INTELLIGENT TUTORING SYSTEMS
12. AI TECHNIQUES, HARDWARE, SOFTWARE, AND NEURAL NETWORKS
13. LOGISTICS FOR AI APPLICATION

Table 1. Artificial Intelligence Technologies Most Relevant to Army Needs

Along with the plan, a definition was necessary. The AMC AI Master Plan defines Artificial Intelligence as "..computer software that exhibits characteristics which when exhibited by humans, will be recognized as intelligent".

Impact of AI on Army Systems

The AI technologies were considered for impact on Required Technical Capabilities for all systems and operations in the Army's battlefield mission areas (including modernization), system development, logistics, and training. The process involved in the review is depicted in Figure 1. As indicated, training is identified as a function that

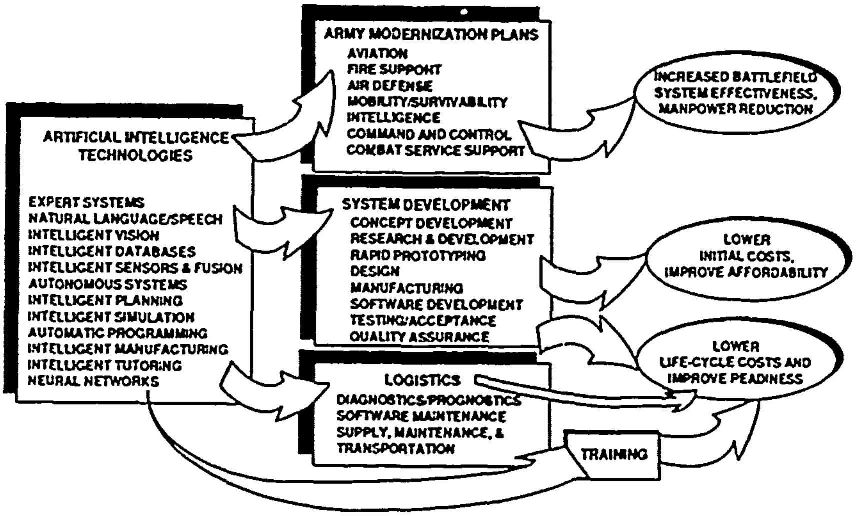


Figure 1. Impact of AI Technologies on Army Missions

impacts all areas of the Army missions including system effectiveness, affordability, and readiness.

A second element in the Master Plan strategy is to distribute the research and development of the AI technologies among the AMC laboratories and research centers. Interested organizations were invited to submit a proposed program of activities, including background on previous AI technology research and development efforts. Teams were formed to pursue technology development in the identified AI technology areas. A technologist from each team was selected to perform as an AMC technology leader. AMC player groups and AMC leaders have been estabished for each of the technologies listed in Table 1. A major objective and responsibility of the AMC technology leader is to plan and prepare for coordinating and leveraging efforts ongoing in other commands, universities, national laboratories, and agencies in the civilian sector.

Technologists from the U.S. Army Missile Command Redstone Arsenal, AL were designated as AMC leaders in the technology areas of Intelligent Tutoring Systems (ITS) and Neural Networks. The focus of the remainder of this report is on the AMC Master Plan for Intelligent Tutoring Systems.

INTELLIGENT TUTORING SYSTEM MASTER PLAN

The vision for the Intelligent Tutoring System Master Plan is to advance and apply the state-of-the-art in ITS to meet the training needs of the Army into the 21st Century. Having stated this, we need to identify, with minimum explanation, a justification for the application of AI to education, tutoring, and training in support of traditional teaching and training methods.

The 2-Sigma Problem

A recent study by B. S. Bloom confirmed the effectiveness of tutoring in small groups with an expert tutor.[4] Using average performances of the control group, results from this research indicates that a teacher presenting material to 20-100 students is one of the least effective methods for educational delivery. Improved results are obtained when the expert teacher not only gives a lecture but involves diagnostic tests to identify where the student might have problems and misconceptions with the subject matter. Student performance in this educational setting is in the 84th percentile compared to the traditional trained student of 50-60th percentile. The most significant results of this study are the results that show the students involved in one-on-one tutoring performs at the 98th percentile or 2-Sigma beyond the conventional teaching. These results from Bloom's study are depicted graphically in Figure 2.

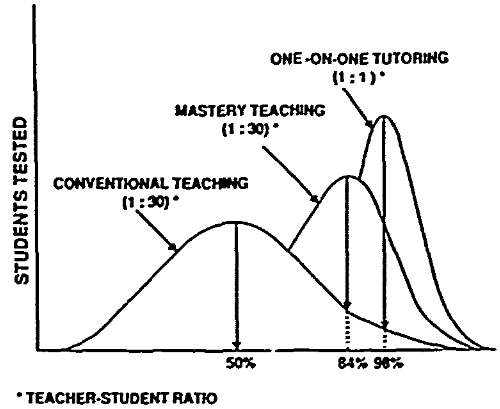


Figure 2. Advantage of One-to-One Tutoring (Bloom 1984)

The results from the Bloom study further confirms the need for a system for education that will facilitate achieving the effectiveness that can be accomplished with a one-to-one human expert tutor/teachers. The resource intensive nature of one-on-one human tutoring precludes this approach from being used except for very special circumstances. What is needed is a device or technology based system that emulates the capabilities and characteristics of a human tutor with wide scale availability and suitability.

Intelligent Tutoring System

During the 1960's the use of the computer in the educational environment was the computer assisted instruction (CAI) system. The goal of the CAI approach to computer use was to produce and present the learning environment with explicit control of the student's learning. While CAI system grew to operate in very complex teaching environments, the challenge of achieving the intended goal was more than could be handled with the basic CAI approach.

Early uses of AI techniques in CAI operation were called generative CAI systems since they stressed the ability to generate problems from large data bases representing the subject they taught. Reactive learning CAI produced an environment in which the student is actively engaged with the instructional system and his interest and misunderstandings drive the tutorial dialogue. With increased emphases on knowledge-based operations and more complex models of the student evolved to drive the system operation, Intelligent CAI (ICAI) became the focus of development efforts.

With the increased use of multiple knowledge bases, cognitive models for reasoning about the student understanding of the domain being taught, ICAI advanced to a threshold identified as intelligent tutoring systems (ITS). Intelligent Tutoring Systems, edited by Sleeman and Brown[5] include papers identifying emerging concepts that provide foundations that drive much of the present day research in ITS. The general trend of ITS theory and development during the past decade is indicated in: Foundations of Intelligent Systems, edited by Polson and Richards[6], Intelligent Tutoring Systems, Lessons Learned, edited by Psotka, etc.[7], and Intelligent Tutoring Systems, Evolutions in Design, edited by Burns, etc.[8].

Intelligent Tutoring System Technology Development

Limited but realizable educational goals have been achieved with ITS in specific areas of training and skill transfer using presently available technologies, see Woolf[9]. The combined technologies of artificial intelligence including: learning models developed in cognitive science; software and symbolic computing architectures developed in Computer Science; the increased cost-benefit ratio of modern microchip based computing are direct contributors in developing currently available intelligent tutoring systems. Additional details on ITS technology base needs will be described in other sections of this paper.

Present activities in the use of AI in the development and application of ITS can be viewed as having two fundamental but different thrusts; one, the focus on the use of ITS with education, including basic principles for grade school and/or college; second, the use of the ITS principles to provide training with skill and knowledge sustaining for a particular domain.

While these two approaches share common principles, the thrust of ITS architectural development places major emphasis on different functional operations. An example, one difference can be viewed as to the degree that emphasis is placed on the transfer of knowledge and the transfer of skills. Skills and knowledge are not independent entities but the degree of emphasis on each entity and the particular performance environment produces different system structures. Intelligent tutoring systems used in applications in an educational environment, which generally has a focus on knowledge transfer, the element of time, in the sense of real-time, are rarely considered. Intelligent Tutoring Systems with a focus on knowledge and skill sustaining, reasoning about problems with real-time applications are critical elements in the learning environment. An example indicating the challenge in teaching real-time tactical thinking is reported by Ritter and Feurzieg.[10]

There are no clearly defined boundaries in the application of ITS for training versus tutoring. While education pays attention to both skills and knowledge, the literature frequently uses the terms training and tutoring interchangeably. As our understanding of what is required in the transference of expertise from expert to novice, and teaching higher order reasoning skills, the boundary between training and tutoring will become less definable. Where it may be deemed appropriate and for definitional purposes here, the application of an ITS for tutoring versus training will be characterized by the degree of modeling as to how the human solves the problem. As will be shown in other sections, this focus on modeling human problem solving increases the reliance on cognitive modeling and requires a flexible architecture for implementing the top level ITS modules.

ANATOMY OF AN INTELLIGENT TUTORING SYSTEM

The anatomy of an ITS can be characterized as consisting of a combination of top level modules. For purposes of focusing on research issues, researchers have identified ITS top level modules to include: human-machine interface, instructional module; diagnostic module; and the domain expert module. Each module is characterized and expanded to include functions that are shaped by special applications and interests of the researcher. The functional structure of the top level modules are depicted in Figure 3.

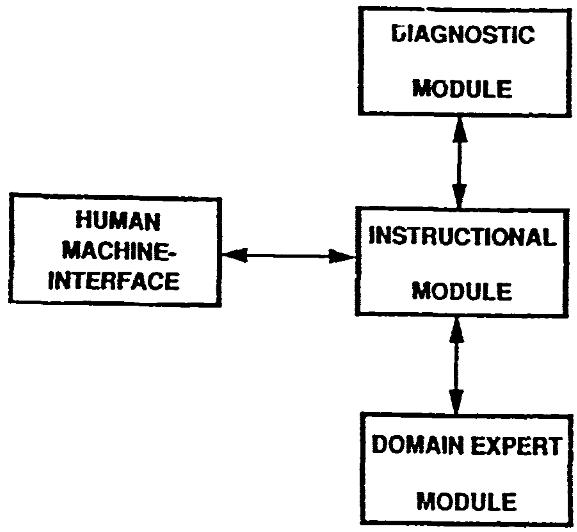


Figure 3. Intelligent Tutoring System Top Level Modules

The transfer of complex knowledge and skills to a student requires an extremely flexible intelligent human-machine interface (IHMI). The IHMI is an integral part of a user conscious system and must maintain knowledge about the user, which includes. what are the preferred methods of interaction; what are the interest, values, and goals of the student; what are the expectation and assumptions of the user? This includes a user dependent knowledge base or user model that would be shared with other ITS knowledge operations. Many factors have been identified for knowledge based systems not being effectively used outside the laboratory and development environment. The human-machine interface has been identified as the most consistent source responsible for rejection of the system by the user.

The instructional module (IM) generates the environment that emphasizes conceptual understanding, levels of abstraction and concept fidelity appropriate to provide the learner with motivation. This module includes the ability to reason about appropriate tutoring strategies for achieving effective and efficient learning for a particular student with individualized learning objectives. An integral part of the IM is the generations of appropriate domain knowledge which could include the use of intelligent simulations. Simulations are used to support task generation and presentation to the student in support of curriculum and instruction operations.

The diagnostic module (DM) has a focus on developing a model of the student or learner's current state of knowledge. Effective tutoring requires the student response to an instruction be compared to the domain expert's response. Differences are analyzed and deficiencies are identified with appropriate knowledge generated for adding to a knowledge structure, or student model, that reflects the user's current state of knowledge about the domain. The student model knowledge is used to identify tutoring strategies, curriculum and instructions to satisfy individual learning objectives of the particular student. It is this feature that gives ITS a unique advantage over other computerized learning systems.

Capturing and encoding the domain knowledge in the expert module (EM) is one of the major tasks in developing ITS. The knowledge in the domain expert module is one of several knowledge bases that can have common usage in an ITS. The expert modules include an expert system with special features that can be necessary for ITS operation. Different types of knowledge, i.e., procedural, declarative, and qualitative, dictate instruction strategies. Options should be available for knowledge representation and reasoning about that knowledge. Constructing an architecture for the expert domains required for ITS operation remains a major challenge in ITS development.

While the ITS can be viewed as modular structured for purposes of research and development, the heart and soul of future ITS will be characterized by the seamless operation of: qualitative reasoning and planning, integrated operation of distributed knowledge bases, and learning and discovery environments for student motivation.

ENABLING TECHNOLOGIES FOR ITS DEVELOPMENT

The ITS can be viewed as a modular structured system with top level modules as identified above. ITS can also be viewed as a tool for educational purposes for teaching basic principles and supplying encyclopedic knowledge bases for the student. Research and development during the past two decades have produced a number of intelligent tutoring systems for specialized domains. Lessons learned from Army sponsored research are providing undeniable indicators that ITS technology trends are moving toward developing training systems with specialized domains of operation (see References 7, 11, and 12). Results from these demonstration models are twofold: first, that effective ITS in limited domains can be constructed; second, progress in developing effective ITS requires advancements and building on past accomplishments in both theory and technology on broad fronts. The enabling technologies include high speed knowledge computing, speech understanding and speech recognition, real-time knowledge based systems, semiautomatic knowledge acquisition and knowledge representation methods, intelligent simulations, intelligent planning, intelligent communication between parallel knowledge systems, and architectures for modular structures operation.

The use of enabling technologies is critically important for present system implementation and future systems development. This implies a close coordination and active support for technologies that are now identified as emerging and technology groups identified and characterized as AI subtechnology areas. This also includes supporting some technology areas not directly identified in the present AI technology base, i.e., cognitive modeling. The complementary nature of enabling technologies that require continuing development for present and future ITS are indicated in Figure 4.

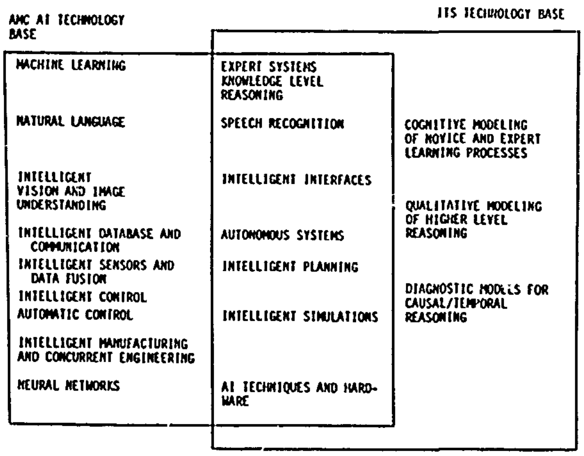


Figure 4. Intelligent Tutoring System Technology Base

VISION FOR AMC AI INTELLIGENCE TUTORING SYSTEMS

The goal of the Master Plan is to promote and advance the development and application of Intelligent Tutoring Systems to meet the Army needs. Three objectives are identified in support of this goal: advance the state-of-the-art in ITS; apply the state-of-the-art in ITS; and identify and implement major thrusts to further facilitate advancing and applying the state-of-the-art in ITS. The functional requirements within the established objectives are identified as: near term, 1 to 2 years; mid-term, 3 to 7 years; long term, and 8 to 17 years. A summary of the near, mid, and long term activities identified in the ITS Master Plan for meeting Army training needs are shown in Table 2.

	ADVANCING STATE-OF-THE-ART IN AREAS OF:	APPLYING STATE-OF-THE-ART AREAS OF:
NEAR TERM	GENERIC ITS ARCHITECTURE FOR INDIVIDUAL AND CREW TRAINING TOOLS FOR REAL-TIME ITS IMPLEMENTATIONS MODELING FOR TEMPORAL REASONING AND REASONING UNDER CERTAINTY COGNITIVE MODELING FOR TEACHING, LEARNING AND CAUSAL REASONING	ANALYZE ARMY TRAINING REQUIREMENTS PROTOTYPE DEVELOPMENT OF ITS WITH GENERIC ARCHITECTURE FOR INDIVIDUAL TRAINING VERIFICATION AND VALIDATION OF ITS OPERATIONS
MID-TERM	SHELL ARCHITECTURE DEVELOPMENT FOR REAL-TIME ITS APPLICATIONS MODULAR GENERIC ARCHITECTURE FOR FORCE LEVEL TRAINING SYSTEMS ADAPTIVE HUMAN-MACHINE INTERFACE USING STUDENT-BASED SENSOR DATA COGNITIVE MODEL DEVELOPMENT FOR KNOWLEDGE AND SKILL TRANSFER	DEVELOP ITS TECHNOLOGY DEMONSTRATION FOR INDIVIDUALIZED OPERATOR TRAINING SYSTEM DEVELOP ARCHITECTURE FOR ITS OPERATING AS AN EMBEDDED TRAINING DEVICE
LONG TERM	MICRO-CHIP TECHNOLOGY FOR ITS DELIVERY PLATFORM THEORY AND ARCHITECTURE DEVELOPMENT FOR REAL-TIME ITS WITH MULTIPLE AND DISTRIBUTED KNOWLEDGE BASES COGNITIVE MODELING FOR DIAGNOSTIC EVALUATION OF STUDENT'S MENTAL MODEL, MISCONCEPTS AND REASONING	DISTRIBUTED ITS FOR TEAM TRAINING WITH CREW MEANDER GEOGRAPHICALLY DISPLACED DEVELOP PORTABLE, PERSONALIZED ITS ADAPTABLE TO SPECIALIZED DOMAINS WITH MODULAR COMPONENTS FOR DOMAIN KNOWLEDGE

Table 2. Research and Development Activities for ITS Development

MAJOR TECHNOLOGY THRUST IN ADVANCING ITS DEVELOPMENT

Confirming cost-benefit ratios and assessing risk factors is a major issue in technology development programs. The major technology thrust program for ITS is a development effort that will demonstrate the confluence of AI technologies in advancing ITS development. The focus of the program is to enhance the development and implementing the functionality of the top level modules described in a previous section. The ITS technology thrust program includes three elements: an ITS architecture development effort for implementing and integrating the top level ITS modules to meet Army training needs; a shell for use in a development environment which includes developing and implementing ITS functions; a micro-chip based ITS for accomplishing symbolic/numeric processing in a delivery environment. A summary of the major elements in the major thrust in advancing ITS development is indicated in Table 3.

ARCHITECTURE
• ANALYZE ARMY TRAINING REQUIREMENTS
• KNOWLEDGE ACQUISITION AND REPRESENTATION IN MULTIPLE DOMAINS
• REAL-TIME OPERATIONS
• COGNITIVE MODELING
SHELL
• ITS PROGRAM DEVELOPMENT ENVIRONMENT
• INTEGRATION OF ITS FUNCTIONS
• INTELLIGENT MAN-MACHINE-INTERFACE
MICRO-CHIPs
• SYMBOLIC/NUMERICAL PROCESSORS
• ARCHITECTURES TO SUPPORT DELIVERY ENVIRONMENT
• MAN-MACHINE-INTERFACE
• REAL-TIME ARCHITECTURE

Table 3. Major Thrust in ITS Development

THE VISION CONTINUES

The development and application of AI technologies to ITS development has been an ongoing effort at MICOM Research, Development, and Engineering Center since 1984. An early cost-benefit analysis was conducted for developing an ITS for a major fielded air defense system. Returns on investment (ROI) were estimated at 7.3. This ROI figure is significant only when based on the degree of realism injected in the evaluator's viewpoints. Returns on investment were based on an economic life of

15 years and manpower reductions associated with training and maintaining critical skills for 100 fielded units. An unspoken assumption in this ROI calculation was that human expert trainees/tutors were available to accomplish the training task and the use of ITS reduced the manpower required to train the student. A more realistic situation is to consider skills that are required but not adequately taught in an institutional or fielded environment due to the lack of the availability of human expert trainers/tutors.[13] An ITS approach would not totally correct this situation, but would significantly improve the conditions. With this viewpoint included in cost benefit calculations, the ROI would increase significantly. This additional factor, however, requires answering the question "What is the cost of not being trained?"

This brief overview of the AMC AI Master Plan also includes a more detailed look at a program for advancing the technology of intelligent tutoring systems. Presently, this is a continuing effort in the planning stage and not readily subject to a conclusion, so the vision continues. The AMC ITS Master Plan is specifically directed toward advancing the state of the ITS technology to satisfy Army training needs. Developing ITS is worth the investment if this technology found application only in this domain.

The need to address the 2-Sigma problem extends beyond the boundary of military needs. The need for an effective teaching system is urgently needed at all levels of our population. If we accept reports about the state of education in our society -- education is in trouble. The slow decline in achievement scores is indicative of the need for a new approach in education. A fully developed ideal intelligent tutoring system will not solve all the problems of education and training. However, in the process of developing such devices, most interesting and exciting challenges lie ahead. These are but a few of the fascinating opportunities provided by these new machines.

ABOUT THE AUTHORS

Willard M. Holmes is a Senior Research Scientist in the Army Missile Command's Research, Development, and Engineering Center, Redstone Arsenal, AL. He leads research and development efforts in real-time simulations and applying AI technologies to the development of tactical and smart missile systems. He has a BEE from Auburn University, MSEE from Oklahoma State University, and D.Sc. from Southeastern Institute of Technology.

Mr. Admiral Piper is a Research Electronics Engineer within the Engineering Management of Instrumentation/Tactical Engagements Simulation Branch at PM TRADE in Orlando, FL. Since 1985, he has been PM TRADE's technical focal point for Robotics. Recently he directed two exploratory development efforts with applications to the National Training Center, Ft. Erwin, CA, for reducing crew size of surrogate combat training vehicles operating in formation. He holds MSc degrees in Electrical Engineering, Engineering Mechanics, and Mathematics from the University of Alabama - Tuscaloosa, and is a member of IEEE, Eta Kappa Nu and Sigma Xi.

REFERENCES

1. AMC AI Master Plan, (Draft) April 26, 1991.

2. Feigenbaum, Edward; McCorduck, Pamela; Nii, H Penny; The Rise of the Expert Company, Times Books, 1988.

3. Fields, C., "The Strategic Computing Program: An Overview," Signal, August 1984.

4. Bloom, B. S., "The 2 Sigma Problem: The Search for Methods of Group Instruction as Effective as One-to-One Tutoring," Educational Researcher, 13:4-16, 1984.

5. Intellient Tutoring Systems, Sleeman, E., and Brown, J. S., (Eds.) Academic Press, 1982.

6. Polson, Martha C., and Richardson, J. Jeffery, (Eds.), "Foundations of Intelligent Tutoring Systems," Lawrence Erlbaum Associates, Inc., 1988.

7. Psotka, Joseph, Massey, L. Dan, and Mutter, Sharon A., (Eds.), "Intelligent Tutoring Systems, Lessons Learned," Lawrence Erlbaum Associates, Inc., Hillsdale, NJ 1988.

8. Burns, Hugh, Partlett, James W., and Redfield, Carol Luckhardt, (Eds.), "Intelligent Tutoring Systems, Evolutions in Design," Lawrence Erlbaum Associates, Inc., Hillsdale, NJ 1991.

9. Woolf, Beverly, Intelligent Tutoring Systems: A Survey in Exploring Artificial Intelligence, Shroe, Howard E., (Ed) Morgan Kaufraahm Publishing, Inc., San Mateo, CA 1988.

10. Ritter, Frank, and Feurzieg, Wallace, "Teaching Real-Time Tactical Thinking," Intelligent Tutoring Systems, Lesson Learned, Psotka, J., Massey, D., and Mutter, Sharon A., (Eds.) Lawrence Erlbaum Associates, Inc., Hillsdale, NJ 1988, pp. 285-301.

11. Holmes, Willard M., and Chamberlain, David, "A Scenario Generator for an Intelligent Tutoring Device Used in Maintaining Operator Skill Levels," Proceedings of the 1988 Summer Computer Simulation Conference, Seattle, Washington, The Society for Computer Simulation International, San Diego, CA, July 25-28, 1988, pp. 613-619.

12. Torre, Jr. James P., Maxey, Sr. Jeffery L., and Piper, Admiral S., "A Validation of Rifle Marksmanship Simulation," Proceedings of the 1988 Summer Computer Simulation Conference, Seattle, Washington, The Society for Computer Simulation International, San Diego, CA, July 25-28, 1988, pp. 653-657.

13. U.S. Army Infantry School, "Training Device Letter Requirement for HAWK Institutional Training System," Directorate of Training and Doctrine, New Systems Training Office, Ft. Bliss, TX, August 1985.

A Generic Model for Rapid Estimation of CBT Development Time

Kurt W. Miles
Applied Science Associates, Inc.
P.O. Station 18
Oklahoma City, Oklahoma

A review of the literature on the amount of time it takes to develop an hour of computer based training (CBT) reveals that figures range from less than 50 hours of development per hour to over 800 hours of development per hour of CBT instruction. An Instructional Systems Design (ISD) model is presented and related to a CBT development estimation model to provide a framework for the discussion of rapid estimation of CBT development time. The estimation model discusses some of the variables that affect development time and offers a method of estimating CBT development time using an simple Job Aid (included) that can be modified to meet local conditions and parameters.

INTRODUCTION

A review of the literature on the amount of time it takes to develop an hour of CBT quickly reveals two facts. first, most of the articles and books that discuss estimating the development of CBT don't provide numbers; and second, the ones which do provide numbers give such a wide range of numbers to choose from that they disagree even within the stated range. The range here is significant, figures range from less than 50 hours of development per hour to over 1000 hours of development per hour of CBT instruction.

It is clear that these authors are using different elements upon which to base their computations, even though they use the same wording to describe their results. The purpose of this paper is to propose some ideas for creating a common ground for discussion of the topic of "How long does it take to create an hour of CBT?"

All the authors caveat their presentations with the statement that the complexity of the CBT to be produced will greatly affect the time it takes to produce the CBT. Other qualifiers which are brought to the attention of the reader are such things as the experience of the subject matter experts (SMEs), writers, and developers; what the customer wants the program to do; and a host of other variables that, justifiably, make giving a hard and fast figure a dangerous prospect. In addition, the factors that the various estimators take into account and the heuristics used to come to a conclusion reveal that different developers include different elements of the ISD process to produce the final figure.

This paper offers a generic model and a Job Aid to help improve the accuracy of the estimate of development time. This model has been used by the author on several projects, with a high degree of agreement between the estimated and actual time. In addition, each iteration further refines the accuracy of the estimation process for this specific environment. Precise figures, unfortunately, verge on proprietary information.

In order to provide some common point of reference, the ISD process used in this article will consist of five elements, analysis, design, development, implementation, and maintenance. Some models include updates as a separate step. This model will include the update as part of the maintenance step.

Analysis compresses the front end analysis of the problem or task, delineating and defining the problem, and reducing it to a form that may be addressed in detail. Design is the actual strategy formulation to resolve or address the problem, and includes planning for the testing to ensure that the proposed strategy will work. Development is the process of creating the structure which will be used in the implementation stage to produce the CBT. Storyboards are an example of a development step. These steps, for the most part, comprise the pre-production portion of a project. Implementation is the act of putting the proposed solution or strategy into action. Many developers see this as a production step, accomplished by programmers, and may not include the time used here in the estimate of time to develop the CBT.

Evaluation of the instruction is inherent in the concept of instructional system design, and is ongoing throughout the process, including implementation and maintenance. It is not addressed as a specific step, but is assumed to be present in all steps of CBT development. The last step is the maintenance of the CBT solution. Many vendors regard this as a separate contract item. Most commonly, it appears to be left out of the time estimate for CBT development.

PROBLEM

When providing a figure for the development of an hour of CBT, developers often fail to specify which of the five steps are being included in the quotation. Thus, one quotation for 150 hours of development may not include the implementation stage, while a quotation for 210 hours may include the implementation, but leave out part of the analysis phase, which was done in preparation for bidding a contract. Which one is a better deal for the customer?

Dr. Robert Spears (personal communication, Nov. 29, 1990) of Simms Industries, Inc., a courseware production company, states that he does not include the implementation in his estimate of time to develop CBT because the work is not developmental. The developmental work is done by the instructional designers and analysts in the first three phases. Their work is fed into a computer program that generates standardized frames. In phase four, implementation, programmers are giving attention to exception frames, frames that need some special logic or branching work. The programmers are not designing instruction, they are producing it.

This brings up another variable. How do you account for the differences in the programming or authoring systems? A developer working in BASIC will have a very different estimate of time from a developer working with QUEST 3.0 or the WICAT authoring system.

Another important factor in determining the time it takes to develop an hour of CBT is the level of the CBT product. There are a number of "standards" in print; anywhere from three to five levels seems to be common. To provide a common reference point in this area, the definitions that appear in the Request for Technical Proposal (RTP) for Navy Contract N61339-87-R-2043 will be used. This RTP has the advantage of specifications that are readily observable and which represent a reasonable coverage of the types of CBT that are being considered in this paper. The definitions are attached at ANNEX A.

LITERATURE REVIEW

* Gery (1986) presents a series of variables that affect the time it takes to develop one hour of CBT [5]. Gery also presents a series of charts that illustrate how those variables interact, and allows a prediction for the amount of time it takes to produce one hour of CBT. The range devised by Gery is from 85 to 300+ hours for an hour of CBT.

It should be noted that the hour of CBT referred to in Gery's article is defined as "an hour of instruction at the computer taking a course that is essentially linear in nature, includes conditional feedback and restricts the use of conditional branching to review comments" (p. 37). This is roughly equivalent to a level 1 CBT, according to the definitions used in this paper.

Gery states "This definition represents the bulk of courseware currently under development. Truly conditionally branched instructional materials are few because of their development complexity and the time limits that most developers have to complete assignments" (p. 37).

* Bork (1985, p. 144-145) illustrates an estimation of development time for a 10 hour CBT project in his discussion of the development process [1]. His estimate for the project was one and one-half years. Calculating the times listed for each development step, it would require 422 5 hours to develop one hour of the CBT. Bork does not specify the level or type of CBT being designed; however, the audience and time frame for when the book was published would imply that it is likely to have been a mid-level 1 to mid-level 2 CBT.

* Air Force Pamphlet (AFP) 50-58 (1978) gives an estimate of 150 - 300 hours of development time per contact hour of CBT [13]. Again, looking at the CBT which was reasonably available, this would likely have corresponded to a mid- to high-level 1 CBT.

* The Navy document referred to earlier provides a useful definition for level 1, 2 and 3 CBT. It also provides some historical data indicating the expected development time for an hour of CBT instruction. The Navy baseline figures for CBT instructions are.

- Level 1 (Baseline Representation - 450 hours)

- Level 2 (Medium Simulation Presentation - 620 hours) and

- Level 3 (High Fidelity Representation - 800 hours) CBT.

* Lee & Zemke (1987) surveyed a number of leaders in the CBT development field as well as published documents, and reported the following data [7].

- C Jackson, a technical director, US Army Armor School, Ft. Knox, KY, uses a 300:1 ratio for estimating CBT, but notes "time increases with sophistication, i.e., simulations require more development time than tutorial or drill- and-practice" (p. 76). He also looks at the nature of the learning objectives, characteristics of the trainees and the capabilities of the staff.

- Other developers at Ft. Knox use ranges as low as 160:1 to as high as 500.1 for CBT development.

- Dewey Crib, president of Instructional Science and Development, a San Diego based CBT vendor, looks at a series of variables (see Gery, above), but quotes the "industry 'standard' for CBT without video is 300-400 hours [per hour of instruction]." Also keep in mind that he is talking about hours of development time for an experienced crew of CBT authors and designers (p. 77).

- Greg Kearsley, CEO of Park Row Software, estimates a CBT development hour's ratio of 200.1 to be ample, but gives the actual range of hours used as 50-500 hours to 1 hour of instruction.

- In 1972, the Office of Personnel Management produced a Training Cost Model document, where the development time for CBT was up to 350 hours for an hour of student contact. (Looking at the CBT capability which was reasonably available, this would most likely have been a level 1 CBT.)

* Soulier (1988) states that "to develop one hour of interactive computer based material requires an estimated 100 to 500 hours [12]." Soulier's model includes phases one through three, as well as producing the user manual and documentation.

* Dean & Whitlock (1988) give a figure of 100 - 200 hours per hour of instruction [4]. However, they ask what is included in the production figures, adding that the whole process from initial discussions through validation and production of other supporting materials is likely to lead to a figure nearer 200:1, or higher.

* In a study of contract requirements for a courseware development program, Miles (1990) found that the figure used to estimate the time required to develop one hour of CBT (without interactive video) was approximately 380 hours [10]. Adding interactive video to the CBT raised the time estimate to almost 475 hours for each hour of CBT. A weighted average of the two types of CBT gave an estimating average of about 450 hours per hour of CBT.

* In an evaluation study conducted by Dawson & Miles (1990) after the program had started, it was found that the number of hours required to produce an hour of CBT was very close to 430 hours, not including support personnel [3]. Adding in a company standard value for support personnel, the actual time to produce the on-line lessons was close to 450 hours. The variables involved in the development project made this a reasonable figure for the product; it also indicates that the courseware development program's estimate was quite accurate.

* Kearsley (1983) states that the commonly used rule of thumb is 200 hours of development for an hour of CBT [6]. He also notes that the time required will vary considerably with the type of CBT, the capabilities of the system, and the experience of the personnel. He later addresses Avner's work, and shows a range of 6 - 610 hours of development time, depending on the instructional structure of the material, and the experience of the authors.

* Mikos, Sullivan, Hebein & Casey (1987) conduct-

ed a study of a professionals in the training development field participating in a workshop during an NSPI Convention [9]. They were asked to estimate the time it would take to develop a specified CBT product. Their answers were then compared to the time the project actually required. The authors' findings indicate that 20 of 27 participants missed the actual time by more than 20 percent. The participants that used heuristics to aid them in the estimation process gave widely varying figures, from 40:1 to 200:1.

* Casey, Mikos, Sullivan & Hebein (1988) repeated the experiment at the following year's NSPI Convention [2]. This follow-on experiment resulted in a finding that 14 of 19 participants missed the actual time by more than 25 percent. In both of these studies, the formally trained developers tended to have more accurate answers than those who learned by on the job experience or self teaching.

ESTIMATION MODEL

How long does it take to develop an hour of CBT instructional development? To echo several of the authors above, "it depends."

If the estimator takes into account all the variables listed by Gery, Kearsley, Dean & Whitlock, Casey et al., and applies common sense and personal experience, they may still miss the mark. Not by their own fault, but because of factors beyond their control which affect the time of development. Customer requirement changes, missing data, personnel turbulence, equipment malfunctions, or just bad guessing can affect the estimate.

Gery suggests that using a system similar to her charts and variables will help an estimator become more accurate than just guessing. While not promising absolute numbers, her charts do lead one into at least considering these variables and their affect on the development process, leaving the estimator with some ballpark figures for comparison.

Casey, et al., provides the reader with a series of estimating worksheets, and some ideas on adjusting the baseline figures that are first used to estimate an "ideal" project. Mikos, et al., list some factors to be considered when adjusting the baseline. Numbers here are a bit more specific, but if the original estimates used to generate the baseline are off, the final results will be off. As indicated above, most of the subjects in the study were off. Clearly, the area needing the most help is devising the initial figures.

What should a developer do to estimate a development project? There are several steps in the process, but one of the most important steps is to do some research on the project to be bid.

Obtain a copy of Gery's variables and look at the logic she outlines. If the time and information are available, make a list and address the questions that you can't answer to the client. Take what you learn and try to plot it on a graph similar to Gery's. Don't attach numbers yet, just get an idea of where the final point on the chart should be, in regards to high or low development costs.

Many organizations have detailed worksheets and specific cost guidelines for developing instruction. Arthur Andersen has 11 different categories, or schools, each of which has a specific cost associated with the development effort (Lee & Zemke, 1987) [7]. If your organization has such information or standards, then much of the fuzziness associated with the estimation process will already have been taken care of.

Unfortunately, many companies don't have the experience, historical data or mental set required to have accumulated this type of information. And many of those that do have this information regard it a proprietary data and are reluctant to share it with outsiders. In this case, the estimator has their work cut out for them.

What variables should be used, and where do they come from?

As we have seen, the published research varies widely in range. It is obvious that each estimate used has been adjusted or configured to fit a specific model that applies to the person making the bid. The most common range offered is 200-300 hours per hour of CBT. An off-the-cuff, informal estimate from a developer will usually fall in this range as well.

The articles that address or use these numbers tend to agree that this is what we would call Level 1 CBT. The sources imply, but do not state, that these numbers include what we are calling phases 1 - 4 (analysis through implementation). No one seems to state whether this includes support tasks (secretaries, librarians, system support personnel, etc.) or not, but the assumption appears to be that it does include the time for these personnel as well.

Pull out the specifications for the program. Determine (if possible) the level of CBT you are trying to develop, what type of authoring system you are using, the number and level of expertise of your people; and what other unusual requirements may exist.

Determine what steps in the development process are to be included in the bid. It makes a large difference if the production and the analysis phases are excluded.

As indicated previously, there are five steps in the development model being used. The fifth step, maintenance, is usually a long term prospect with a relatively low manpower or time requirement. Unless the project is an in-house program, a reasonable approach is to make the maintenance phase a separate contract, and charge it that way, thus excluding it from the initial bid.

Establish consistency between bids.

To establish some consistency from bid to bid, or in development quotes, this author believes that it is a wise idea to plan the estimate according to phase of development, and then to present the estimate (at least internally) as a set of four numbers, one for analysis, one for design, one for development and one for implementation. After arriving at these four numbers, the estimator can eliminate those phases that are not to be included in the final figures. When comparing estimates, it becomes a matter of comparing development phase vs. development phase (phase three vs. phase three) rather than development time vs. development time (phases 1-4 vs. phases 2-3).

This does not appear to be what is happening in most cases. The reader should not infer that it is not happening internally within the companies that develop courseware After all, profit is the difference between what it costs and what is charged. That is one reason many companies regard the numbers and processes they use as proprietary data Accurate estimation takes a lot of work.

Quick estimates.

What about the student or individual who needs to make a quick, rough estimate of a development effort? They should first find out, at least in general terms, what level of complexity the courseware will have. Anything

above a level 1 CBT will increase the time. The higher the level, the more time needed

Ask about the system being used to develop the courseware. If it is a relatively sophisticated and standard system such as WISE, QUEST, REGENCY, etc., then not much adjustment is needed. Just about anything else will require some modifying of the estimate, usually upward. Programming in BASIC, or Pascal would be expected to take the longest.

Think about the people. What is their experience level in each phase? How many of them are there? Are they co-located? Have they worked together long? Any working environment where the personnel are not a long-term, experienced, co-located team will drive the estimate upward.

Look at the availability of existing material. If this is a brand new course, and subject matter experts are few or documentation is sparse, development time will go up. Is there a standard development procedure? If not, count on an increase in time while one is developed, formally or informally.

Allocate time.

Now look at how you will divide the time up between the phases. Generally, any work done in the analysis phase will have significant impact on the amount of work that needs to be done in the following phases. Thirty to fifty percent of the effort could be spent in the analysis phase.

If the analysis work was well done, then 15 - 20 percent of the effort can be spent in the design phase.

Twenty-five to forty percent of the effort is spent in developing the instructional strategies, instructional frames, and sequencing. This compares very closely to the results of Sampath & Quaine (1990).

The remaining time (from 10 - 30 percent) goes to production.

As you can tell by adding up the numbers, you could spend from 90 to 140 percent of the time you have. Caution is advised when cutting your estimates too close.

Allocating 40 percent of the time to analysis, 20 percent to design, 25 percent to development and 15 percent to production will account for 100 percent of the time. Using the "industry standard" common estimate of 300 hours for an hour of Level 1 CBT allocates the time this way:

Analysis	120 hours
Design	60 hours
Development	75 hours
Implementation	45 hours

For a perfect world and a perfect project, this can give you a set of factors to use. If you don't have to account for implementing (programming) the program, subtract 45 hours If the analysis has been done, and done well, subtract 120 hours. Not so difficult, after all.

However, as we don't live in a perfect world with perfect projects, I would like to present a model of estimation to adjust those perfect numbers.

Use the Job Aid to make adjustments.

Figure 1 (CBT Development Estimation Aid) has the questions talked about earlier listed with a marking scale. It also has the phases listed across the top.

Feel free to change the values across the bottom if you have other numbers you believe are more accurate.

Add up the number of hours per phase, and you have an estimate (based on 300 hours) of what it will take to do the program.

Look at phases 1 and 2. There are two blocks are blacked out, since the analysis and design do not relate to the level of the CBT or the system on which they will be produced. At that point, hopefully the analysis team is still open to the medium to be used. In any case, the level and type of system to be used are not critical.

Each tick mark is equal to one-tenth of the value of the phase. If you believe that the project you are estimating deserves more or less tick marks than shown on the chart, feel free to place them in. This is only a rough guide, after all As you use it, you may find that you need to adjust the hours up or down from 300, or to shift hours from one phase to another.

Then, subtract those phases you are not responsible for, and give your estimate.

Summary.

How accurate is this job aid? It gives you a more accurate estimate than "somewhere between 8 and 1000 hours per hour." And it takes into account at least some of the variables that affect courseware development. It helps to quantify the steps, and gives you a tool that will let you compare numbers with some level of confidence.

Later, when you have time to do a more detailed study, you can compare your estimate with Gery's charts and see how close you are. You can also compare your estimate with actual program time, and adjust your estimating figures. The more you do, the more accurate your projection will be. As always, experience plays a key role This job aid will, hopefully, help you gain positive experiences more quickly than you otherwise would, on your own.

Figures 2 and 3 are some examples of more refined checklists, one of which is a LOTUS 1-2-3 template.

To answer the question asked at the beginning - how long will it take - we don't know, for sure. But we can make an educated estimate.

REFERENCES

[1] Bork, A. (1985). Personal computers for education. New York: Harper & Row.

[2] Casey, R. J., Jr., Mikos, R., Sullivan, D. J. & Hebein, J. M., (1988). Capturing skill in estimating training development effort: A follow-up final report. Performance and Instruction, 27(9), 40-44.

[3] Dawson, R. G. & Miles, K. W. (1990). [Courseware development progress and status]. Unpublished raw data.

[4] Dean, C., & Whitlock, Q. (1988). A handbook of computer based training (2nd ed.). London. Kogan Page.

[5] Gery, G. J. (1986). How long will it really take?: Estimating CBT development costs. Data Training. July, 34-37.

[6] Kearsley, G. (1983). Computer-based training, a guide to selection and implementation. Reading, MA. Addison-Wesley.

[7] Lee, C., & Zemke, R. (1987). How long does it take? Training. June, 75-80.

[8] McDonnell Douglas Corporation. (1988). Proposal 88D-001 (Phase II). St. Louis: Author.

[9] Mikos, R., Sullivan, D. J., Hebein, J. M., & Casey, R. J., Jr. (1987). Estimation training efforts/costs: A follow-up "final report" (part 1). Performance and Instruction, 26(5), 25-29.

[10] Miles, K. W. (1990). CBT product definition of the C-17 Aircrew Training System. Unpublished manuscript.

[11] Sampath, S. & Quaine, A. (1990) Journal of Computer-Based Instruction. "Effective Interface tools for CAI authors." 17(1), 31-34.

[12] Soulier, J. S. (1988). The design and development of computer based instruction. Boston: Allyn and Bacon.

[13] U. S. Air Force. (1978). Handbook for designers of instructional systems (Air Force Pamphlet 50-58, Volumes I - VI). Washington, DC: U.S. Government Printing Office.

ABOUT THE AUTHOR

Kurt Miles received his bachelor's degree in Psychology from Texas A&M University in 1975. After graduation, he entered the US Army as an active duty artillery officer, where he served for twelve and one-half years. He left the service to earn his Master of Education degree in Instructional Technology from Utah State University, and is currently working on his Ph.D. In March of 1991, he accepted a position with Applied Science Associates, Inc. as an Instructional System Designer. He previously worked for Simms Industries, Inc. as a Test and Evaluation Specialist. Kurt is a member of the Federal Educational Technology Association (FETA) and is active in the Association for Educational Communications and Technology (AECT).

The CBT Estimation Job Aid is available from the author as LOTUS 1-2-3 template. Please contact the author if you are interested in obtaining the template. An alternate address to one listed at the beginning of this paper is:

Kurt Miles
3805 Highpoint Ct.
Norman, OK 73072

ANNEX A
Information from
Request for Technical Proposal (RTP)
for Navy Contract N61339-87-R-2034.

Level 1 - Baseline Presentation:

This is the lowest level of interactive courseware (ICW) development. It is basically a knowledge or familiarization lesson, in linear format (one idea after another), used mainly for introducing an idea or concept. The trainee has no control of what is seen (minimum trainee activity). The two (2) types of Baseline Presentation are:

A. Video and minor text presentation

Information/knowledge type lessons

Simple questioning techniques

Low trainee interactivity (basically page turning)

No multi-tasking required

Limited branching

Rudimentary remediation

Real time events presented with location type video using the videodisc media

No real time simulation

Minor graphics/text overlay on video (i.e., headings, captions)

No mathematically driven modeling required.

B Graphics and minor text presentation (No video)

Computer generated graphics (CGG) presentation

Information/knowledge type lessons

Predominantly simple text and simple CGG pictures

Simple questioning techniques

Low trainee interactivity (basically page turning)

No multi-tasking required

Rudimentary remediation

Use of magnetic tape(s), floppy disk(s), or videodisc media

No real time simulation

Restricted to simple geometric animations, text

No mathematically driven modeling required

The historical data provided in this RTP shows the expected number of hours of development time for an hour of CBT instruction at Level 1 to be an average of 450 hours.

Level 2 - Medium Simulation Presentation

This medium presentation level involves the recall of more information than a baseline Level 1 presentation and allows the trainee to have increased control over the lesson presentation (i.e., touch screen or light pen to rotate switch). A moderate degree of simulation is used in the presentation. This presentation shall provide the following:

Combined information and skill lessons

Moderate degree of programming

Trainee interactivity with various I/O devices

Computer Managed Instruction(CMI) to track and analyze student performance

Normally combines video and graphic presentation.

The historical data provided in this RTP shows the expected number of hours of development time for an hour of CBT instruction at Level 2 to be an average of 620 hours.

Level 3 - High Fidelity Presentation

Primarily used for procedural task/skills

High student interactivity

Extensive branching capability (falls short of artificial intelligence)

Maximum remediation opportunity (i.e., multiple responses measure degree of error and give relevant responses)

Real time simulation with minor equipment limitations (i.e., timing sequences of start-up, switch changes)

Capability to interface with other output devices

Exhaustive CMI capability

The historical data provided in this RTP shows the expected number of hours of development time for an hour of CBT instruction at Level 3 to be an average of 800 hours.

CBT Development Estimation Job Aid

Question	Phase 1 Analyze	Phase 2 Design	Phase 3 Develop	Phase 4 Produce
<u>CBT Level</u> Level 1 √ Level 2 √√ Level 3 √√√√				
<u>Development System</u> Sophisticated author system √ Limited authoring system √√ Authoring language √√√ Programmed √√√√				
<u>People</u> Inexperienced √ New team (<1 yr) √ Separated √ 2 or 3 of above √				
<u>Other Factors</u> Existing materials yes some √ no √√ Existing standards yes no √				
Number of marks ---->				
Multiplied by ---->	12	6	7.5	4.5
Equals ---->				
Plus ---->	120	60	75	45
Hours per phase ---->				

Based on 300 hrs of development per hour of Instruction. Hours allocated 40% Analyze, 20% Design, 25% Develop, 15 % Produce. Base hours and percentages should be adjusted to meet your specific requirements, when known.

Figure 1
CBT Development Estimation Aid

Quick CBT Estimation Job Aid

Question	Phase 1 Analyze	Phase 2 Design	Question	Phase 3 Develop	Phase 4 Implement
<u>Task Complexity</u> Simple -1 Average 0 Complex +2 Highly Complex +4			<u>CBT Level</u> Level 1 0 Level 2 1 Level 3 3		
			<u>Development System</u> Sophisticated Author System -1 Authoring System 0 Authoring Language 2 Program Language 4		
<u>Personnel</u> Inexperienced 1 New Team (<1 Year) 1 Old Team (>2 Year) -1 Separate Locations 1 2 or more above 1					
<u>Other Factors</u> Existing Materials Yes -1 Some 0 No 1 Existing Standards Yes -1 No 1					
Score--------------->					
Multiply by -------->	12	6		7.5	4.5
Equals ------------->					
Plus --------------->	120	60		75	45
Equals Hours per Phase					

Based on 300 hrs of development per hour of Instruction. Hours allocated 40% Analyze, 20% Design, 25% Develop, 15 % Produce. Base hours and percentages should be adjusted to meet your specific requirements, when known

Figure 2
CBT Development Estimation Aid

	Phase 1 2 3 4	Phase 1 Analyze	Phase 2 Design	Phase 3 Develop	Phase 4 Implement
CBT Level					
Level 1	0 ----0 0				
Level 2	1 ----0 0				
Level 3	3 ----0 0				
Task Complexity					
Simple	-1 0 0 ----				
Average	0 0 0 ----				
Complex	2 0 0 ----				
Highly Complex	4 0 0 ----				
Development System					
Sophisticated Author Sys	-1 ----0 0				
Authoring System	0 ----0 0				
Authoring Language	2 ----0 0				
Programming Language	4 ----0 0				
Personnel					
Inexperienced	1 0 0 0 0				
New Team (< 1 year)	1 0 0 0 0				
Experienced Team (>2 yr)	-1 0 0 0 0				
Separate Locations	1 0 0 0 0				
2 or 3 of above	1 x x x x				
Other Factors					
Existing Materials					
Yes	-1 0 0 0 0				
Some	0 0 0 0 0				
No	1 0 0 0 0				
Existing Standards					
Yes	-1 0 0 0 0				
No	1 0 0 0 0				
Number of Tick Marks --->		0	0	0	0
Multiplied by --->		12	6	7.5	4.5
Equals --->		0	0	0	0
Plus --->		120	60	75	45
Hours per phase --->		120	60	75	45
Estimate of all phases --------------------------------------->					300
Base Hours & Percent per Phase		40.00%	20.00%	25.00%	15.00%

300	Hours Base	Recommendations	
40.00%	Analysis	40.00%	300 Hrs Level 1
20.00%	Design	20.00%	610 Hrs Level 2
25.00%	Develop	25.00%	820 Hrs Level 3
15.00%	Implement	15.00%	
100.00%	Total Percentage		

Figure 3
CBT Development Estimation Aid

TRAINER TEST AND EVALUATION PROCESS REVIEW

CDR Paul S. Kenney, Paul R. Little, and R. Thomas Galloway

Naval Training Systems Center

Orlando, Florida

ABSTRACT

This paper presents the results of a process review of the Naval Training Systems Center development test and evaluation procedures used in the majority of its current contracts. Data were derived from a survey of project engineers, 79 completed contracts, interviews with 11 simulator manufacturers, and contacts with the National Simulation Evaluation Program (FAA) and local Defense Contract Management Command Area Office (DCMAO). Recommendations are made for improved test planning, changes to the Contractor Preliminary Inspection process, interfaces to MIL-STD-2167A and general policy guidelines for test policy and practices.

INTRODUCTION

Training systems, in many cases, represent the most complex systems procured by the Navy. They are somewhat unique when compared to other systems: specifications are performance based detailed hybrids; usually procured on a single or very low production basis; the prototype is the first "production" unit; trainer design begins after or concurrent with the parent system, however, the Ready-For-Training date often precedes delivery of the parent system; the time available for testing becomes compressed; man-machine considerations (behavioral and ergonomic) are important; and early operational data availability for trainer use is a problem. In addition, trainers mostly use commercial hardware components and are software intensive. This combination of characteristics has spawned a test and evaluation philosophy that differs from operational systems. This philosophy may or may not be optimum in today's environment.

The Naval Training Systems Center (NAVTRASYSCEN) process of testing a training device is a serial sequence of test processes beginning with a preliminary test by the Contractor followed by the Government, and a final test phase at the training site first by the Contractor then by the Government. Government acceptance occurs upon successful completion of this sequence of serial tests. Specification and Statement of Work (SOW) language for these tests have been, with small adjustments, unchanged for 17 years. The contract language used in the SH-2F Helicopter Weapon System trainer contract dated 15 May 1973 is virtually identical to the current specification language. The current NAVTRASYSCEN device testing process is illustrated by Figure 1.

The key features of this figure are as follows:

a. The test process begins after critical design review (CDR) approval.

b. Navy Preliminary evaluation (NPE) defined by MIL-D-8708B, may be held at the earliest possible opportunity to determine: (a) potential or existing deficiencies of the trainer; (b) to highlight the need for identification and early correction of deficiencies and (c) to evaluate changes incorporated. NPEs have been used in aircraft simulators to verify the flight dynamics early in the development cycle, and in the surface program as a mini Test Readiness Review.

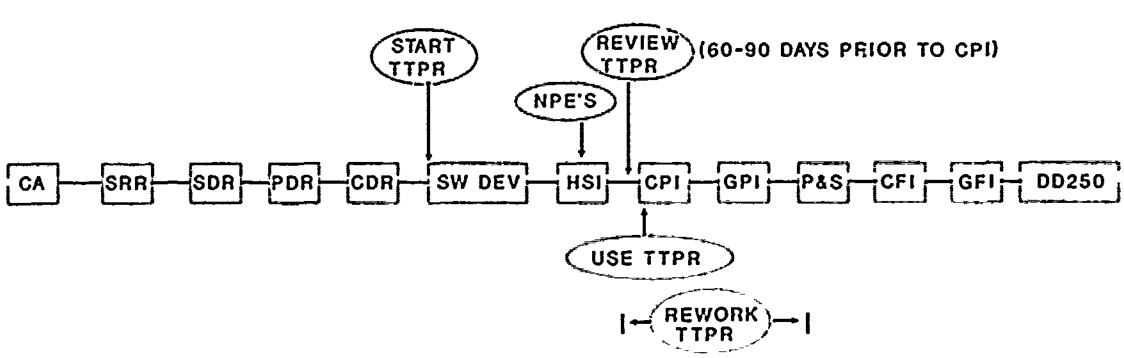


FIGURE 1. CURRENT T&E PROCESS

c. The Contractor develops the draft Trainer Test Procedures Report (TTPR) document prior to the start of Contractor Preliminary Inspection (CPI).

d. The TTPR (as a draft) is submitted for review and comment 60-90 days prior to the CPI.

e. Upon Government authorization, CPI is commenced by the Contractor with Government monitoring.

f. Upon completion of CPI, identified discrepancies are corrected and the TTPR updated.

g. Government Preliminary Inspection (GPI) is commenced and includes Functional Configuration Audit (FCA), Physical Configuration Audit (PCA), software cold start, and the execution of the revised TTPR under Government-controlled test conditions. Discrepancies are formally identified and the trainer is under configuration control. Tests include functionality tests and mission operability tests.

h. Upon completion of GPI, identified discrepancies are corrected, verified, and the training device is shipped and installed at the training site.

i. The Contractor's Final Inspection (CFI) verifies successful reassembly at the training site and successful implementation of final corrections.

j. Government Final Inspection (GFI) reruns the TTPR (usually selected portions), including cold start and extensive mission tests to ensure final operability and implementation of the specified performance.

k. GFI includes functional tests of the equipment operations through mission tests and in some cases unconstrained missions which executes the widest latitude of trainer functionality.

l. Upon successful completion of these tests, device acceptance is executed.

m. The Trainer Test Procedure Report is now finalized with the result of the tests and becomes the Trainer Test Procedure Results Report (TTPR/R). This document is subsequently used to verify baseline performance of the training devices.

DATA SOURCES

Data were derived both internal and external to NAVTRASYSCEN. External data collection was from:

o Unstructured interviews with 11 simulator Contractors.

o The National Simulator Evaluation Program, Federal Aviation Administration.

o Defense Contract Management Command Area Office, Orlando, Florida.

o Review of Air Force Systems Command Trainer System SPO YW Operation Instructions (Reference 1).

o Naval Air Test Center (References 2 & 3) publications.

o Visit to DELTA Airlines Simulation Facility.

<u>Interview with Industry</u>

Representative members of the simulation industry were invited to meet with the test and evaluation committee. These meetings were unstructured and no agenda was provided by the Government other than our interest to appreciate the test problems as viewed by our Contractors. The focus was to have a constructive dialogue and receive recommendations which may be offered. The 11 Contractors who participated in this dialogue represented approximately 67% of the total dollars for all contracts awarded in FY 90 and were as follows:

1. Loral
2. E&S
3. Link CAE (TSD)
4. Reflectone
5. McDonnell-Douglas
6. Quintron
7. Grumman ESD
8. Hughes (HSSI)
9. General Electric
10. AAI
11. Lockheed Sanders

The National Simulator Evaluation Program managed by the Federal Aviation Administration (FAA), Flight Standard Division, was visited by members of the committee. This visit focused on understanding the National Simulator Evaluation Program and the recent proposed Airplane Simulators Advisory Circular, AC No. 120-40B, Reference 4, and the Airplane Flight Training Device Qualification (Draft) AC No. 120-45A, Reference 5. The extent of overlap between the FAA simulator standard development process and the NAVTRASYSCEN simulator contract process was explored through discussions of lessons learned and items of mutual concerns and benefits.

The Defense Contract Management Command Area Office (DCMAO) develops and monitors procedures for process control, test, and inspections to meet contract designated requirements. Should DCMAO be asked to inspect or qualify the functionality of equipment operation, the degree to which this could be supported is related to the skills and background of the current employees. In most cases, Weapon System functionality exhibited by the system under test could be witnessed as to occurring and under what conditions,

but the goodness or nuances of partial performance would not be directly detected. Under current practice DCMAO usually requests NAVTRASYSCEN to provide engineers or subject matter experts with the necessary performance knowledge.

Internal data were collected from:

o Review of 79 completed contracts over the last 6 years.

o Direct survey of NAVTRASYSCEN Project Engineers.

o Review of prior studies, reports, and current practice.

DISCUSSION

The following discussion summarizes the major process observations based on the review of previous data sources. The test and evaluation committee sought to highlight major significant items. Several items of data are thought to be secondary indications of primary events.

<u>Trainer Test Procedure Results Report (TTPR)</u>

For 27% to 29% of our contracts, the TTPR is not acceptable at the start of CPI. In addition, for new first article simulators, an average of 12% of the time the TTPR is unacceptable at the start of GFI. This is illustrated in Figure 2.

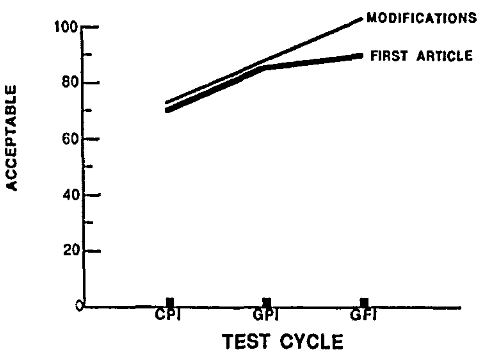


FIGURE 2. TTPRS ACCEPTABLE AT START OF TEST CYCLE

For the unacceptable TTPR documents, the degree of completion at CPI was 60% for first article, rising to 84% by GPI, and again dropping to 78% for GFI. This is illustrated in Figure 3. The Government assumption that the TTPR would be virtually complete at the beginning of each serial test is obviously erroneous.

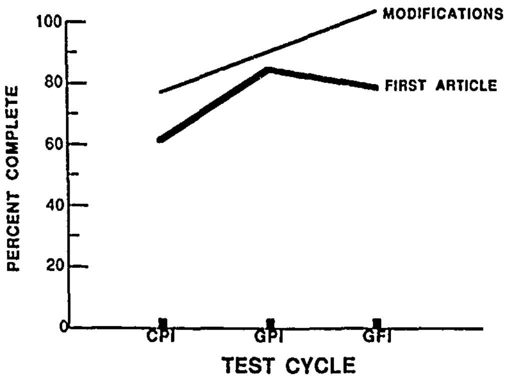


FIGURE 3. TTPR PERCENT COMPLETE

The second assumption of our serial test model is that early identification of discrepancies during CPI would diminish the number of subsequent discrepancies during further testing. The average number of discrepancy reports observed during each test cycle is shown in Figure 4. It is clear from the rise of discrepancy reports (DR's) during the GPI test cycle that CPI is not a completed process by the beginning of GPI.

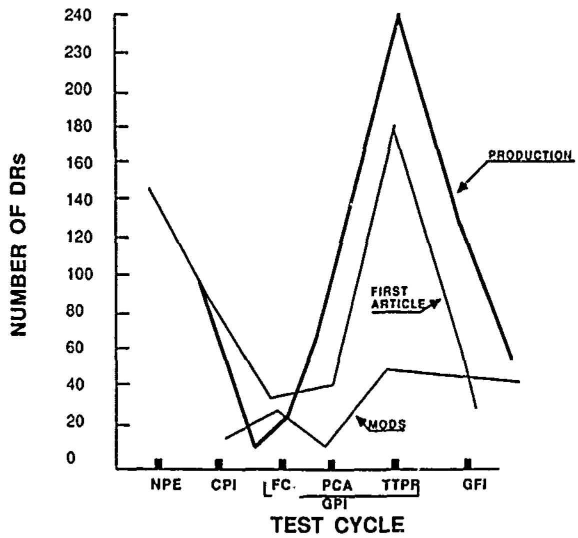


FIGURE 4. DISCREPANCY REPORTS BY TEST CYCLE

Discussions with industry and our engineers have led the testing and evaluation process review team to examine the impact of software. Our current test practice has been identified to be essentially unchanged for over 17 years. During this time, the growth of our simulation software in aviation programs, Figure 5, has grown by an order of magnitude. During this time, simulator manufacturing characteristics have changed primarily from hardware intensive to software intensive. This change to software development under the process of MIL-STD-2167A has not been incorporated successfully into our test practice.

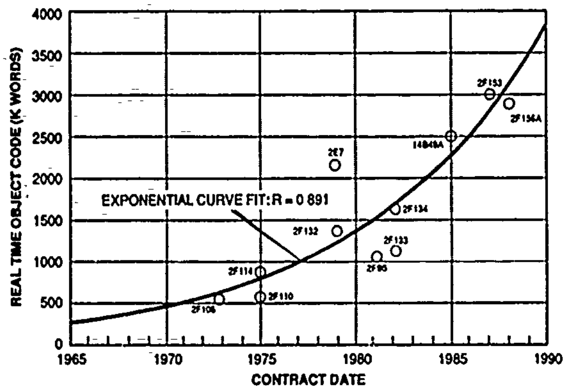


FIGURE 5. GROWTH TRENDS IN SIMULATION SOFTWARE CONTRACT AWARD YEAR VS REAL TIME OBJECT CODE

Computer development standards were reviewed and included DOD-STD-2167A, Reference 6, Defense System Software Development, and Defense System Software Quality Program, DOD-STD-2168, Reference 7. In addition, the tri-service Joint Integrated Avionics Working Group (JIAWG) tailored DOD-STD-2167A, Reference 8, guidance was obtained. This document establishes policy and standards for on-board software or firmware on JIAWG aircraft using DOD-STD-2167A. The goal is to ensure that Computer Software Configuration Items (CSCI) maintain the same DOD-STD-2167A development process across all JIAWG programs. This will affect aircraft simulator software development and testing when interfacing to the JIAWG software.

Computer Software Growths

It appears that hardware/software integration (HSI) is being extended on software intensive simulator developments to overlap the start of CPI and in some cases, the start of GPI. This HSI may be planned but is not consistent with the approved schedule and anticipated test in the SOW. This is seen as incomplete test documents and delays in the start of testing, a symptom very evident in the in-house surveys. For the current NAVTRASYSCEN device testing process as illustrated in Figure 1, the key point is the linkage between HSI tests and the TTPR revision process. These feedback loops contribute to excessive cycling of TTPR revision. On the other hand, the manufacturer uses the hardware/software integration and the Contractor's preliminary inspection time to develop and test the TTPR, and often will carry this over into the Government preliminary inspection process. The trainer consoles become the instruments used to test the software. NAVTRASYSCEN's software SOW's anticipates the testing of software, primarily through equipment operation, which reinforces this model. In highly complex software developments, the linkage of CSCI test and HSI test to total system testing was not identified and not consistently planned from the early initiation of the contract activity.

Results of Industry Meetings

In order to evaluate NAVTRASYSCEN Test and Evaluation (T&E) procedures from the supplier's perspective, eleven 3-4 hour meetings were held with individuals representing various training system Contractors. Based on these open discussions, a pattern of themes emerged. These themes are provided in Figure 6 in perceived priority order. Specific issues raised by industry are presented here as areas of concern and recommendations for improvement from the Contractor's perspective.

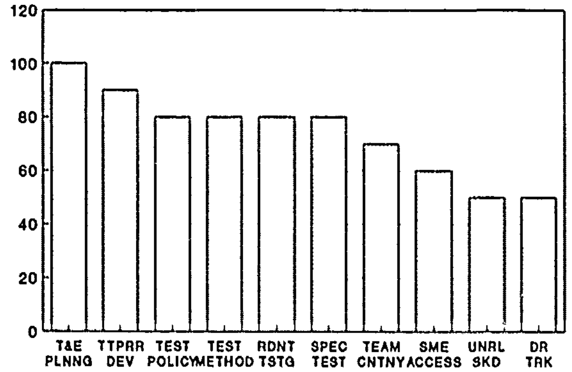


FIGURE 6. MAJOR PROBLEMS IDENTIFIED BY INDUSTRY

a. T&E Planning

NAVTRASYSCEN programs do not always require the development of a program Test and Evaluation Master Plan (TEMP). The lack of a TEMP or equivalent document can result in a disjointed and poorly planned test evolution. As a result, issues can arise during the test phase which can severely impact schedule and the success of the procurement.

Industry Recommendation:

NAVTRASYSCEN requires the development and use of a TEMP to address all areas of T&E as they apply to that specific program. The TEMP should be a contractual document that is jointly prepared by the Contractor and the Government. The TEMP should be a living document that matures as the program develops through design and into HSI (Hardware/Software Integration). A T&E working group consisting of members from NAVTRASYSCEN, Subject Matter Experts, the FPT (Fleet Project Team), and the Contractor should be established as the focal point for all areas concerning T&E.

b. Trainer Test Procedures Report

The current process for TTPR development, submission, and review is incomplete. TTPR submission is usually required 60 days prior to the commencement of CPI and the document is frequently disapproved by the Government. Many rounds of revision and resubmission occur during CPI/GPI and sometimes into CFI/GFI.

NAVTRASYSCEN often requires the TTPR to test each specified performance

parameter of the training device from a given set of initial conditions. Even though many of these parameters could be tested as a group, they are detailed in the TTPR as segmented tests which often require extensive repetition of switch positions to achieve the same test condition.

TTPR's frequently do not contain detailed procedures for mission testing (sometimes called freeplay). As a result, the area of mission testing is not well understood by the Contractor and represents an area of risk during the test phase.

Industry Recommendation:

TTPR generation should commence early in the program (not later than the PDR and should be a joint Contractor/Government effort. TTPR's should address mission testing and should define the specific mission profiles (provided by the government) to be tested by the operators.

c. Test Procedures

Test procedures seem to vary from one program to another. The current structure of training system T&E leaves many unanswered questions which result in various approaches to testing based on the experiences of the personnel involved with the program. Testing is usually not discussed in detail until late in the program and as the test date approaches. NAVTRASYSCEN has not published its test procedures to industry except as contained in individual RFP's.

Industry Recommendation:

NAVTRASYSCEN define and publish its test procedures for training systems.

d. Test Methods

Certain aspects of training systems have more than one accepted test method for determining specification compliance. This is especially true in visual systems. The test method selected can often effect the test results achieved and hence the systems compliance with the specification. This is also true in the area of aerodynamic testing when comparing automatic test methods to manual test procedures.

Industry Recommendation:

NAVTRASYSCEN determine and publish accepted test methods for the various areas of training system performance. The test methods should be referred to in the RFP. If test methods are not published, the RFP should specify the method to be used in either the statement of work or the detailed specification.

e. Redundant Testing

Current NAVTRASYSCEN contracts allow the Government to require a full running of the TTPR at CPI, GPI, CFI, and GFI. Certain sections of the TTPR could be run only once during CPI, with adequate Contractor quality assurance certification, and not be repeated during subsequent Government/Contractor testing. Repeating all tests on subsequent lots of the same device are also unnecessary and should not be required. Airline flight simulator programs sometimes limit testing of subsequent lots to on-site testing only.

Industry Recommendation:

Early test planning should be accomplished to determine which sections of the TTPR are applicable for the various phases of trainer development and testing.

f. Specifications

NAVTRASYSCEN contracts frequently require the TTPR to address each paragraph of the specification. The Contractor is then required to design a specific test to demonstrate compliance with the specification on a paragraph by paragraph basis. Some performance specifications are general in nature and do not lend themselves to a structured test evolution.

Industry Recommendation:

NAVTRASYSCEN perform a test validity review on specified performance parameters to ensure that compliance can be demonstrated with a structured test.

g. Team Continuity

The lack of Government test team continuity results in frequent changes in the direction and priority of test evolutions. Areas of subjective testing are especially vulnerable to differences of opinion as test team membership changes.

Industry Recommendation:

Although it is recognized that test team turnovers are inevitable, they should be minimized as much as possible. Upfront planning for personnel replacements could help to reduce the disruption caused when critical personnel leave the program during specific test evolutions.

h. Subject Matter Experts

Limited access to Subject Matter Experts (SME's) until the program enters the testing phase often results in misunderstandings between the Government and the Contractor on the relative importance of system performance parameters especially in subjective areas. A better understanding of the Weapon System mission and its tactical employment could enhance the training system design approach and maximize the utility of the system in achieving the desired training objectives.

Industry Recommendation:

SME's be made available to the Contractor early in the program design phase in order to increase the Contractor's familiarity with the simulated weapons system and its operation. Increased availability of SME's immediately after HSI may help to detect major software design errors that might otherwise not be discovered until commencement of acceptance testing. A caution here is that SME's should not be used as a source for reporting performance parameters but rather to highlight areas of concern and to familiarize the Contractor with mission scenarios. The training system specification and appropriate technical documentation should serve as the official source for system performance parameters.

Problems arise when trainer performance fails to meet the user's expectations and he doesn't understand that the causes may be due to inherent limitations of the simulator. The services of experienced technical experts from NAVTRASYSCEN are invaluable for mediating and resolving problems in visual systems, aerodynamic modeling, and motion cueing systems by relating user expectations to practical trainer capabilities, thus reducing the potential for adversarial situations in the trainer test and evaluation process.

i. Unrealistic Schedules

NAVTRASYSCEN contracts typically specify 6 weeks for CPI and 6 weeks for GPI regardless of training system complexity. This is also true for subsequent production lots of the same system. In actuality, in-plant testing of the prototype system may take several months while testing of subsequent production lots could be limited to on-site testing only. NAVTRASYSCEN Request for Proposals (RFPs) tend to be very strict when it comes to program schedule and force the bidders to meet the schedule in their proposal or be considered non-compliant with the RFP.

Industry Recommendation:

NAVTRASYSCEN RFP's be less rigid in scheduled test performance between contract award and RFT dates in order to allow the Contractor to bid the program schedule tailored to simulator complexity and intensity of software development.

j. DR Tracking Procedures

Multiple systems exist for tracking DR's during testing. The lack of a standard DR tracking system requires each program to develop its own system or adopt a system used on another program.

Industry Recommendation:

NAVTRASYSCEN adopt and publish a standard PC based DR tracking system for use on all programs.

ANALYSIS

Test Policy

The current structure of NAVTRASYSCEN contracts requires a CPI which essentially duplicates the Government Preliminary Inspection. CPI is usually observed and certified by the local DCMAO representatives. Current contract language requires correction of "all" deficiencies discovered during CPI prior to the commencement of GPI. Depending on the DCMAO representatives involvement with CPI and their understanding of how the trainer operates, correction of discrepancies found during CPI may or may not be allowed until after the full TTPR has been completed. The nature of the discrepancies found during CPI may require re-running portions of the TTPR to ensure that software corrections to the device have not generated problems in another area. Some DCMAO representatives have required substantial retesting of the device to ensure otherwise nondiscrepant areas of the trainer have not been altered by DR corrections. As a result, once the trainer enters CPI, the Contractor's flexibility to correct discrepancies may be severely constrained.

The recommended change to the above process would replace the CPI (as we know it) with a Contractor in-plant test process which is more flexible and places more responsibility for certifying the trainer ready for GPI on the Contractor. Early development and qualification of a Test and Evaluation Master Plan (TEMP) and supporting T&E working group would serve to improve the process. The TTPR structure would be built via T&E working group meetings and current status reported during the same time frame as progress reviews. As the TTPR develops, it will be reviewed by this team and approved incrementally. As soon as appropriate sections of the TTPR have been deemed "suitable for testing" by the T&E working group, the Contractor would be free to complete those sections of the TTPR at Contractor discretion. The completion of the test and the test results would be certified by the Contractor's quality assurance (QA) department and presented to the Government during follow-on T&E working group sessions.

As the trainer development progresses, sections of the TTPR would fall into one of four categories:

a. Test procedure under development

b. Test procedure ready for review

c. Test procedure approved and ready for testing

d. Test procedure completed and QA certified

As trainer construction and HSI matures, the T&E working group would determine when the device was ready for a Test Readiness Review (TRR). During the TRR, the Project Engineer, assisted by the fleet project team, would run sample demonstration mission scenarios (defined in the TTPR) to verify device readiness for GPI.

Test Methods

Potential candidates for standardized test methodologies include motion platforms, g seats, control loading systems, transport delay/cue synch measurements, basic visual system performance, and basic flight characteristics. Further study may reveal other candidates. The FAA has applied this concept of standardized testing to the qualification process for airline pilot training simulators. Advisory circulars issued by the FAA (References 4 and 5) describe test standards and, to a limited extent, test methods for the candidate areas mentioned above. These advisory circulars clarify the FAA's expectations in advance with respect to the testing required to qualify a simulator for airline pilot training. A similar concept can be established for military pilot training simulators but the scope must accommodate the broader spectrum for military mission tasks. Other resources for standard testing are the flight test manuals published by the test pilot schools which describe the theory and test technique for investigating aircraft performance and handling qualities. The U.S. Naval Test Pilot School (TPS) manuals have been referenced in NAVTRASYSCEN procurements for over ten years. This experience has shown that further documentation is needed to clarify how the TPS methods should be adapted to trainer use and to take advantage of trainer unique features such as parameter freeze and automated test drivers.

In summary, NAVTRASYSCEN should identify candidate test areas and publish standard test procedures for demonstrating training specification performance requirements. Existing FAA and TPS documents should be utilized for guidance in format and content.

NAVTRASYSCEN Project Engineer Survey Results

The adequacy of the specification, SOW, contract schedule, and DD 1423s in the testing area were rated as approximately 3.5 on a scale of 1 (poor) to 5 (excellent). In addition, the survey indicated the statement of work was 2.6, below average, and the contract schedule was 2.9. The major difficulties cited were in the areas of preparing procedures for test, and defining test criteria and fidelity requirements. Government Furnished Equipment (GFE) performances was also cited as a difficulty. While the Technical Proposal Requirement (TPR) may address testing, in most cases, this was not a significant factor in the source selection.

The current test and evaluation procedure and policy is only considered sufficient by 60% of experienced project engineers. There is considerable inconsistency regarding discussion of testing responsibility.

In the area of test and acceptance of training devices the top concerns were as follows:

- Incomplete test procedures at CPI and subsequent schedule impact

- Continuity and skill of Fleet Project Team SMEs

- Contractor indicates trainer is ready for test when it is not

- Poorly written test procedures - GFE operation not understood.

- Unrealistic test schedule - minimum consideration for correcting large number of Discrepancy Reports (DRs).

Major contributions identified for extending the test time were as follows:

- Incomplete and inaccurate test procedures

- Software malfunctions and reliability

- Discrepancy Reports (DR) acceptance and resolution

- TTPR Documentation use incomplete or inaccurate, with limited testing prior to Government test

- Faulty coldstarts, hardware and software failure

- Insufficient time, DR clean-up was slow

A test plan was used by the test team on most cases (the TTPR). The use of DCMAO representatives was minimum, and "free play" test planning was normally not available or coordinated in advance with the Contractor. The test time reported by the engineering group suggested that delay in starting of testing and actual execution of the tests were from 2.4 to 2.8 times the originally planned test duration. These delays are symptoms of the additional time needed for HSI and time required not previously planned for the correction of extensive discrepancies. One program had a delay of a year and was not included in the data. There was no consistent requirement for a TEMP application and use.

RECOMMENDATIONS

The following recommendations are made in order to improve the efficiency and effectiveness of the trainer testing process:

The general approach is to revise the T&E process to help Contractors better understand the test requirements earlier, to begin preparation of test documents earlier, to reduce Government intrusion during the later stage of HSI, and to allow phased development of the TTPR. The proposed process is illustrated in Figure 7.

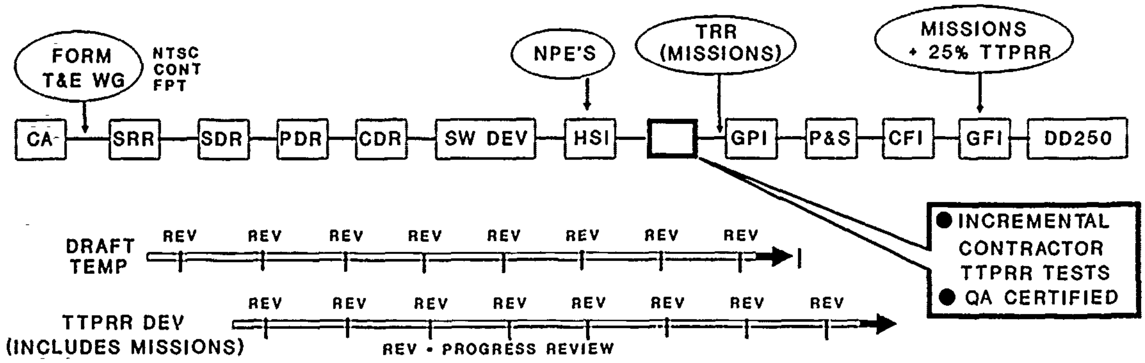


FIGURE 7. PROPOSED T&E PROCESS

o Replace the current CPI with a Contractor controlled process beginning earlier which contain the following features:

a. Use of TEMP, test plans, DR resolution plan

b. Incremental TTPR development including intent of in-process software testing

c. Contractor certification of readiness for Government test

d. Verify readiness via demonstration of Government mission scenarios in a Test Readiness Review.

The following specific process changes are recommended:

Recommendation 1. A Test and Evaluation Master Plan (TEMP) should be developed as a CDRL item and define objectives, critical issues, system characteristics, responsibilities, resources, and schedules for test and evaluation (Reference DoD 5000.3-M-1). It should list the participants and each of their roles. Finally, the plan should state the conditions required for completion of the Test Readiness Review, discrepancy reporting resolution, Conditional Acceptance, and Final Acceptance. A sample TEMP checklist is included in Appendix A.

Recommendation 2. A Test and Evaluation working group consisting of Government project engineers, Contractor, fleet project team and any additionally required SME's should be established. This working group would be responsible for the development and implementation of the TTPR, Test Planning, Test Witnessing, Test Readiness Certification and determination of cold start requirements. This group will report during all progress review meetings on current status and test and evaluation planning for the acquisition, and will resolve differences, develop program solutions, and provide overall direction for the test and evaluation program. This group will document all decisions and agree to procedures relative to the training system test and evaluation. Subsequent Contractor TTPR submission for Government review and approval would not be required. Membership on this Team should be from contract award until RFT in order to ensure continuity throughout the program to the T&E master plan.

Recommendation 3. Incremental Testing during HSI of completed CSCI threads is recommended. This would reduce the current testing redundancy reflected in the manner in which the TTPR is repeatedly run in CPI, GPI, CFI, and finally GFI. Contractor presentations, NAVTRASYSCEN Engineering surveys, and experience within the process review team indicates that TTPR development and implementation could be accomplished so that, with early planning, test sequences could be built in increments and it would not be necessary to repeat many of the tests once they were run and verified. This would allow more detailed test and better confidence than current practice.

Recommendation 4. Mission Scenarios should be provided by the Government for inclusion in the TTPR. These scenarios would be used as the primary system test vehicle, and be identified early in the program. These mission scenarios should be identified in the TEMP document and revisions made when they become known.

Recommendation 5. Form a Joint Industry Government Working Group to improve the focus, structure, and format of the TTPR and Results documentation. The growing software and database content

of training devices will continue. Other process controls, such as DOD-STD-2167A and DOD-STD-2168 will impact the form of data, where it is located, accessed, and verified. The current practice of the TTPR as a self-contained volume can be improved upon.

Recommendation 6. Evidence of satisfactory completion of the TTPR by the Contractor must remain a prerequisite to the Government beginning their tests. This would be accomplished through contractor QA certification followed by a Test Readiness Review (TRR).

Recommendation 7. Change the bid process to allow the bidder to propose the detailed test durations and schedule milestones. In addition, Test and Evaluation planning and process becomes an agenda item for all PDR's, CDR's and Progress Reviews.

Recommendation 8. Discrepancy Report (DR) tracking should be standardized. Included in this standard should be a requirement for the Government to be able to monitor the data base on-line via modem access. NAVTRASYSCEN should develop a model PC database program for use when no contract process applies.

Recommendation 9. Develop, publish, and implement standard T&E policy and procedures for the test and evaluation of all NAVTRASYSCEN developed training systems. This policy and procedures document should apply to all warfare areas and include Development, Test and Evaluation (DTE), and Operational Test and Evaluation (OTE).

Recommendation 10. Automatic Testing routines and procedures should be developed and implemented whenever possible. This feature would be utilized to accelerate the testing process and to ensure trainer life cycle integrity before and after trainer modifications. Such automatic testing should be expanded to be included in the design goals of most trainers.

Recommendation 11. Develop and implement a standard "Memorandum of Agreement" for T&E Programs requiring the participation of DCAMO.

Recommendation 12. Provide NAVTRASYSCEN technical expertise to balance user expectation with contract specification.

Recommendation 13. Support specification items with practical test requirements. The Technical Requirements Specification should be written so that each stated requirements has a corresponding test requirement. This will ensure a better understanding of the requirement and how it will be tested.

Recommendation 14. Develop and publish standard test procedures. Commonly accepted trainer test methods should be available for reference at the beginning of trainer development, preferably with the RFP. These referenced test procedures could serve as standard methods for demonstrating fundamental trainer performance characteristics. These standard methods should also establish a process for developing new or modified test methods to address new or unique trainer characteristics.

Recommendation 15. NAVTRASYSCEN should initiate joint Industry and Government working groups to publish joint test guides of standardized test methodologies.

CONCLUSION

NAVTRASYSCEN is currently relying on T&E practices which were considered effective in the days of analog training devices and in the early years of digital computer training devices. However, unlike past trainers, modern training devices are software intensive and are primarily constructed from commercial off-the-shelf (COTS) hardware. These modern trainers lend themselves to incremental testing of subsystems and mission testing of the entire trainer.

The current T&E trainer process from both external and internal perception is incomplete and does not work well with computer software intensive trainers. Growing software and database complexity will continue and does not fit well in our current serial test model. A proposed change to the CPI process is recommended. This change will replace the current CPI with a Contractor controlled incremental process through early TEMP planning, incremental TTPR developments, incremental testing, and a verification process by a Government mission scenario test readiness review prior to GPI. This will allow better software development and test within the systems performance test structure.

LIST OF REFERENCES

1. Training Device Test and Evaluation (Acquisition Management), YW Operating Instruction 800-7, Department of the Air Force, HQ Aeronautical Systems Division (AFSC) Training Systems SPO, 3 November 1989.

2. NAVAIRTESTCEN Policy for Test and Evaluation of Major Aviation Training Devices, NATCINST 3960.12A, Naval Air Test Center, 10 July 1989.

3. Maximizing Flight Fidelity; Integration of Naval Air Test Center Capabilities into the Procurement of Major Aviation Training Devices, Technical Memorandum TM 76-4 SA, Naval Air Test Center, Patuxent River, 16 March 1977.

4. Airplane Simulator Qualification, FAA Advisory Circular AC No. 120-40B (DRAFT).

5. Airplane Flight Training Device Qualification (Draft) AC No. 120-45A.

6. Military Standard, Defense System Software Development, DOD-STD-2167A, Department of Defense, 29 February 1988.

7. Military Standard, Defense System Software Quality Program, DOD-STD-2168, Department of Defense, 29 April 1988.

8. Tailored DOD-STD-2167A, JIAWG 89-S6/VER: 3.2, Joint Integrated Avionics Working Group (JIAWG), 31 July 1990.

9. Test and Evaluation Master Plan Guide, Department of Defense Instruction 5000.3-M-1.

APPENDIX A

(SAMPLE)

CHECKLIST FOR TEST AND EVALUATION MASTER PLAN (TEMP)

A Test and Evaluation Master Plan defines objectives, critical issues, system characteristics, responsibilities, resources, and schedules for test and evaluation.

---

1. INTRODUCTION

1.1 System Description
1.2 Critical Technical Characteristics

2. OVERALL TEST AND INTEGRATION APPROACH

2.1 Test and Integration Objectives
2.2 Test Classification
2.3 Test and Integration Methodology
2.3.1 Traceability and Compliance
2.3.2 Incremental Builds
2.3.3 Integration Approach
2.3.4 Testing Approach
2.3.5 Critical Items
2.3.6 Regression Testing
2.3.7 Thread Performance Demonstration
2.4 Software Standards and Control
2.5 SIM/STIM
2.6 Coordination and Visibility
2.6.1 Navy-Conducted Tests
2.6.2 Discrepancy Report

3. TRAINER TEST PROCEDURES AND RESULTS REPORT

3.1 TTPR Development Methodology
3.2 Mission Preferences
3.3 Integrating with DOD-STD-2167A Development Test
3.4 Hardware/Software Integration Test Outline
3.5 Trainer Test Procedures Report - Outline
3.6 Special Test Procedures
3.7 Trainer Test Procedures and Results Report - Plan
3.8 Installation and Checkout Plan

4. TEST FACILITIES AND TEST EQUIPMENT

4.1 System Test Facilities and Test Bay Support
4.2 Support Systems (SIM/STIM)
4.2.1 Tactical Operation Interface
4.3 Component Test Equipment and Test Facilities
4.3.1 Software Generation and Test Facility

5. TESTS AND EVALUATIONS

5.1 Critical Items
5.2 Configuration Item Development and Testing
5.2.1 Requirements Traceability
5.2.2 Hardware Unit Testing
5.2.3 Software Testing
5.2.4 Software Cold Starts
5.3 System Integration and Testing
5.3.1 Software Integration and Testing
5.3.2 System Verification and Requirements Testing
5.3.3 System Stress Test
5.3.4 Weapons Compatibility Test
5.3.5 System Software Documentation
5.4 Reviews and Inspections
5.4.1 Reviews
5.4.2 Software Specification Review (SSR)
5.4.3 Preliminary Design Review (PDR)
5.4.4 Critical Design Review (CDR)
5.4.5 Audits (FCA and PCA)
5.4.6 Maintainability Demonstration
5.5 Government and Independent Testing
5.5.1 Preliminary Evaluation
5.5.2 Test Readiness Review
5.5.3 Government and Preliminary Inspection
5.5.4 Government Final Inspection
5.5.5 Subject Matter Experts

6. SYSTEM INTEGRATION AND TEST OPERATIONAL PERFORMANCE

6.1 Mission Summary
6.2 Integration and Test - Plan
6.3 Resources
6.4 System Test Critical Issues
6.5 Test Management
6.5.1 In-Plant Test
6.5.2 On-Site Test

7. SCHEDULES

8. SECURITY CONSIDERATIONS

8.1 Scope
8.2 Unit EMI Test Plan

8.3 Development Tests
8.4 Design Assurance Tests
8.5 Qualification Test Program
8.6 Units Tested
8.7 Test Facilities

9. ENVIRONMENTAL QUALIFICATION TEST PLAN

9.1 Scope
9.2 Applicable Specifications
9.3 Documentation
9.4 Test Approach
9.5 Required Tests
9.6 Test Unit List
9.7 Facilities
9.8 EQT Test Personnel and Organization

10. DISCREPANCY TRACKING SYSTEM
10.1 Introduction
10.2 Documentation
10.3 Discrepancy (DR) Tracking System Implementation

ACKNOWLEDGMENT

The authors wish to thank the other process team members for their substantial ideas and insight; Roger Nutter, Bill Rankin, Al Fliss, Carl Thomas, Paul Brothers, Dan Schab, and contributors John Cheney, Keith Russell, Tony Kuhn, John Mills, and Lori Lane.

The companies from the Simulation Industry who participated provided insight, understanding, and generally a point of view not otherwise available. The candor and preparation was evident and appreciated.

ABOUT THE AUTHORS

CDR PAUL S. KENNEY is assigned as the Project Manager for all A-6E/ EA-6B training systems managed by the Naval Training Systems Center. CDR Kenney is a graduate of the US Naval Academy, U.S Naval Postgraduate School, and US Naval Test Pilot School. He holds both Bachelor and Masters degrees in aeronautical engineering. CDR Kenney has completed two operational tours flying the A-6E Intruder and has accumulated over 3000 flight hours and 500 carrier arrested landings. CDR Kenney has also served as the Operation Test Director for Electronics Warfare Systems while attached to VX-5 and was the Director of Product Support at the Naval Aviation Depot, Norfolk, VA.

PAUL R. LITTLE, P.E., C.P.L., is the current manager of the Surface Warfare Division, and provides for technical development, evaluation, and acquisition of simulator programs in the Surface Warfare area. Earlier assignments at NAVTRASYSCEN include management of System Engineering, Software Engineering, and has directed field offices in San Diego and Pensacola. Mr. Little was a major contributor in NAVTRASYSCEN's improved computer/software acquisition strategies and transition to ADA. He holds both a Bachelor and Master of Engineering Degree in electrical engineering from the University of Virginia in Charlottesville. Mr. Little has 23 years experience in the simulation industry, and is a member of I.E.E.E. and Florida Engineering Society.

R. THOMAS GALLOWAY is an aeronautical engineer and head of the Simulation Dynamics Branch at the Naval Training Systems Center. His responsibilities include the investigation of flight simulation requirements both for rotary wing and fixed wing pilot training. He has participated in the development and evaluation of many Navy flight simulators and has authored several technical papers on flight fidelity test practices. Mr. Galloway holds a BSAE from Georgia Tech, an MSAE from Princeton University, and is a graduate of the US Naval Test Pilot School engineering curriculum. He is a member of the AIAA and the American Helicopter Society.

TODAY'S NEED FOR VIABLE TRAINING MEASURES OF EFFECTIVENESS

Commander David C. Ray, USN
Office of Chief of Naval Operations
Total Force Training and Education Policy Division
Head, Air Training Assessment Section
Washington, D. C.

ABSTRACT

This paper examines the requirement for viable Training Measures of Effectiveness (TMOE). It addresses basic concept considerations for developing useful measures. Today's fiscally constrained environment forces us to defend our training resource proposals by quantifying what is gained in terms of warfighting readiness. While overall warfighting readiness is made up of various components (personnel readiness, training readiness, supply readiness, and material readiness), it is training readiness which is most often equated to our defense capability. Credited as our most potent force multiplier, training is, in fact, the hardest portion of our readiness "make-up" to quantify. This helps explain why training programs are often the first to be axed when pitted against hardware resource requirements. Adding to the problem of grasping training as a hard resource requirement is its complex make up. When examined on a macro scale, warfighting training readiness is broken down into individual, team, unit, and battle group training readiness components. We in the training arena are working to develop measures of individual and team training performance. However, the complexity of measuring unit and battle group operations often dictates the use of expert opinion as the yardstick with which to measure effectiveness. While the merits of this method have satisfied budgeteers in the past, it is now being questioned more and more. The bottom line is that we have yet to discover an acceptable method of linking training to readiness. The successful thwarting of critics who ask, "How much training is enough?", or "What return do the taxpayers get for their dollar?", must be based on quantifiable units of measurement. Baseline assessments of training systems have diminished importance if not based on quality data. The future competitiveness of training versus hardware in the resource arena is based on the need for effective measures of effectiveness.

INTRODUCTION

As the Navy moves into the 1990s an entirely new set of concerns face our decision makers. Advances in technology and an ever increasing gap between recruit entry-level skills and requisite skill levels force an expansion of our training requirements. A simplistic argument may be made that this increase in training requirements will be offset by force structure cuts brought on by a rapidly changing world environment. The truth of the matter is, that projected force structure downsizing will likely yield only horizontal cuts in the training infrastructure. While this affects classroom throughput and the associated support functions, the realized savings are relatively small. With Congress clamoring to cash in on the peace dividend, training resources will, at a minimum, be likely candidates for vigorous Congressional oversight. In 1977 Congress passed Public Law 95-799 requiring the Secretary of Defense to report quantifiable and measurable material readiness requirements. While this law applied only to the material readiness side of the overall readiness equation, it was clear that the "quantifiable and measurable" theme was the underlying, real issue. The question is, "Can this theme be applied to training readiness?" I believe that there is a definite relationship between training support costs and the resultant combat readiness. For over twenty years the Navy has pressed to identify quantified training measures of effectiveness (TMOE) to successfully defend training resource proposals. Although some success has been made on small scale training programs, the need to develop viable TMOE as a decision making tool is real. Today's resource-scarce environment demands the most efficient use of available resources. This paper looks at the complex interaction of variables that have made warfighting analysis and assessment an inexact science to date and offers a potential solution.

BACKGROUND

Training is an essential function in the operation and performance of any branch of the Armed Forces. Within the Navy, this essential function involves a complex network of programs and

requires resources in the billions of dollars. The size and complexity of this training infrastructure demands feedback systems which gauge performance. The Navy views its training readiness system in four basic components; they are Individual, Team, Unit, and Battle Group/Battle Force (BG/BF) Training.

Individual Training

Individual training provides not only knowledge and skills to recruits (officer and enlisted), but also specialized skill training to maintain and operate today's highly sophisticated weapons systems. This specialized training ranges from initial skill training (A School), advanced specialized skill training (C School), and functional skill training (F School) through pilot training to graduate level education. It is estimated that individual training in the Navy will cost approximately 5.5 billion dollars in FY-92. (See Figure 1)

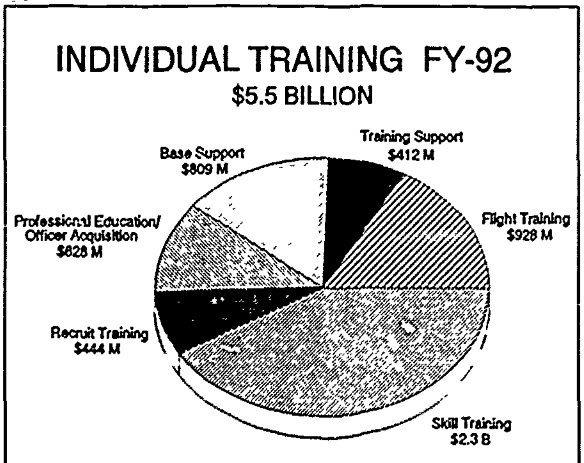


Figure 1

The Navy has been successful in developing effectiveness measures for specific skills. In the Electronic Warfare (EW) rating, for example, the Skills and Knowledge Assessment Tool (SKAT) has been developed. A computer-based test/tutorial, SKAT has shown potential for use as a direct output measure of individual performance. Based on factual knowledge and application of tactical skills, SKAT is a low-cost, minimum impact tool that can provide data upon which warfare area analysts can make qualified recommendations for improvement of training. Other individual training efforts are underway. The Maintenance Training Improvement Program (MTIP) is currently testing all aviation enlisted maintainers in a similar fashion. Whether applied to highly perishable/degradable skills as in SKAT, or high cost programs as in MTIP, this type of performance evaluation and feedback is invaluable for curriculum managers and students alike. Additional efforts are being pursued by all training resource sponsors within the Office of Chief of Naval Operations (OPNAV).

Team Training

Team training encompasses those group tasks and skills which are essential to the mission and operation of a complex unit such as a ship. Successful programs such as the AEGIS Combat Training System (ACTS) employ embedded stimulators for realistic tactical scenarios. The state-of-the-art Versatile Training System (VTS) II feedback system designed within ACTS provides for monitoring and evaluating team training status and effectiveness. These same design features are being used by the Naval Sea Systems Command (NAVSEASYSCOM) to develop an Organic Shipboard Combat Systems Trainer (OSCST) for possible use on older classes of ships.

Unit Training

Unit training is comprised of a series of training exercises designed to accomplish a predetermined level of performance within prescribed warfare mission areas as defined in the individual unit's Required Operational Capability/Projected Operating Environment (ROC/POE). Currently the measuring of ship performance is judged by several methods. The first is a series of exercises at the end of Refresher Training (REFTRA). This series of exercises, graded by a team of observers on each coast, provides a degree of standardization to the grading criterion. REFTRA offers a degree of training measurement in that it can judge a unit's performance improvement over the allotted training period. An additional series of tests comprise a unit's competitive cycle. Within this competitive cycle the unit is graded in such areas as the Operational Propulsion Plant Examination (OPPE). A unit's inspection cycle also offers a period of scrutinizing. In this cycle a unit is inspected for their compliance with the 3-M maintenance system and for the material condition of the unit (INSURV Inspection). While these cycles in themselves provide the Squadron Commander and the individual unit's Commanding Officer a good feel for where they stand in their ability to carry out their assigned missions, nowhere are the results collated and analyzed for training's contribution. In one attempt to do this, a recent Center for Naval Analyses (CNA) study showed a positive relationship between

unit steaming days and unit performance in exercises. While this effort presents a strong argument for protecting quarterly steaming days it does not tie training to specific performance. The complex interaction of individual and team training within a 350 man, multi-mission unit offers an expansive and expensive proposition.

Battle Group/Battle Force (BG/BF) Training

BG/BF training entails a series of exercises designed to "workup" assigned units for extended overseas deployment. These exercises encompass both inport and at-sea training evolutions. Battle Force Inport Training (BFIT) consists of three phases. These phases are designed to exercise organizational, communication, and tactical skills prior to at-sea training periods. Until recently BG/BF at-sea training periods have been an opportunity for BG Commanders to exercise their group under general guidelines set by their respective Fleet Commanders. Over the past few years, Commander Second Fleet has developed standardized Battle Group Operational Readiness Ratings (BGORS) which specifically identify mission training performance criteria. This program sets performance standards for BG Commanders to attain prior to their last Fleet Exercise (FLEETEX) before deployment. Short of actual combat, BGORS, used on existing instrumented ranges, is the best measure of BG/BF training readiness developed to date. Recent joint Fleet Commanders in Chief (FLTCINC) efforts are redefining Battle Group tactical training requirements through the use of a continuum type concept. At this writing, the effort is in it's infancy but promises to provide operational commanders with required levels of performance. Limited range capability, range time, exercise targets, and exercise ordnance all limit the effectiveness of such efforts.

DISCUSSION

Navy training organizational functions can be described in a series of simplified actions which combine to generate the outputs (Training Readiness) for which they were designed. (See Figure 2)

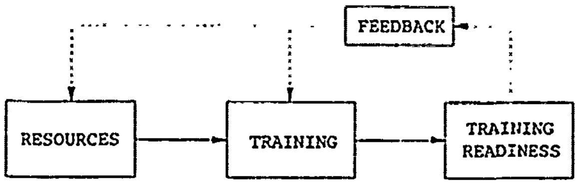


Figure 2

In analyzing the components, we must define the resources that go into the system, how the resources are used, and what output is achieved. Analysis in this fashion can produce feedback to the system. The comparison of the resource costs to the output achieved provides the decision maker information for selecting alternatives which will either maximize the desired output at a specified level of resource use or minimize costs to produce a specific level of output. In today's limited resource environment, either output will produce some level of security risk which if properly articulated would provide necessary budget justification.

Resources

It is estimated that fleet training will cost approximately 12 billion dollars in FY-92. (See Figure 3)

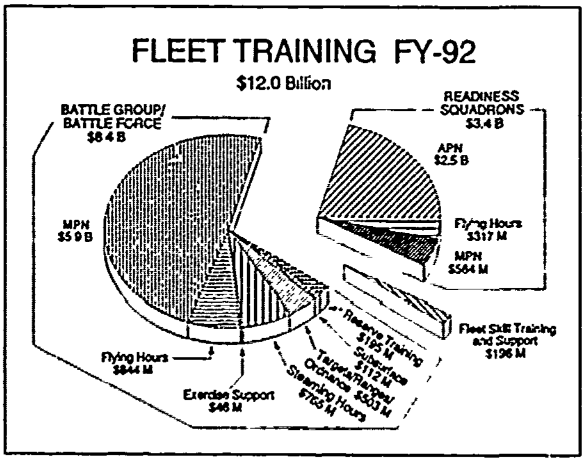


Figure 3

This resource allocation for Navy training represents an annual bill totaling 17.5 billion dollars when the costs of individual training are added. Because training readiness is so important and costs so much, high level defense decision makers understandably want to know how effectively these monies are being spent.

Training

It is beyond the scope of this paper to define in detail the training component of Figure 2. It is sufficient to say that this component is made up of those types of training efforts diagramed in Figure 4.

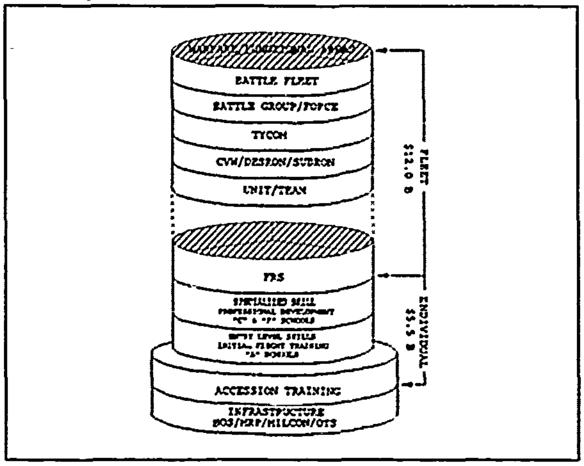


Figure 4

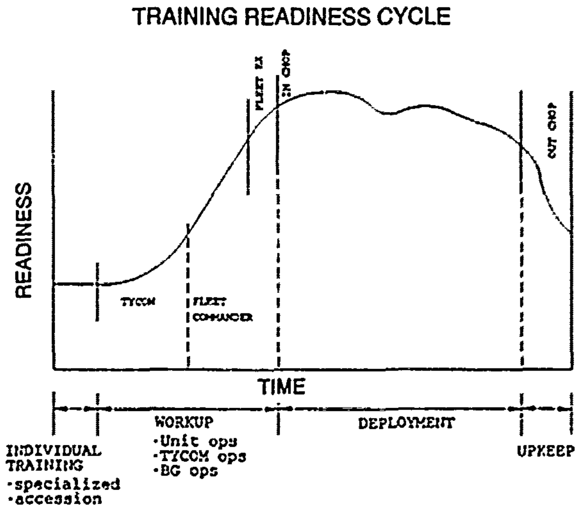


Figure 6

Training Readiness

In order to properly define Training Readiness it is appropriate to first note its importance in the collective makeup of our overall Battle Group capability. Focusing at the unit level, readiness is composed of four distinct readiness components; they are Personnel Readiness, Training Readiness, Supply Readiness, and Material Readiness. (See Figure 5)

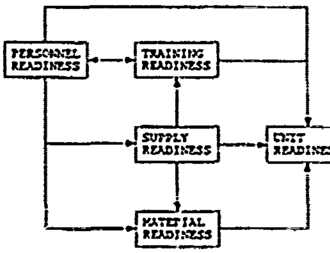


Figure 5

Not to undermine the importance of Supply and Material Readiness, it is Personnel and Training Readiness that are our two most vital contributors to our warfighting capability. Both Supply and Material Readiness can be physically measured. Hard data can be obtained in these areas to justify their importance and impact on unit readiness. Personnel Readiness is based on individual training and the proper care and feeding of the unit's personnel. As discussed in the background portion of this paper, Training Readiness is based on individual training, team training, and unit training.

One commonly accepted description of Navy Training Readiness is depicted in Figure 6. This graph shows the cycle through which a unit's and/or Battle Group's readiness changes throughout its training "workup" period and deployment. The variance in readiness is influenced by many factors. A late start within the cycle or a lack of training services will produce a lower level of readiness at any given period of time. The peak in readiness is designed to occur just as the unit/Battle Group changes operational control (CHOP) to its forward deployed chain of command. Ideally the curve would flatten out at "in chop" and remain constant throughout the deployment. However, experts agree that a dip in readiness occurs approximately half way through deployment. This downward trend is caused by such factors as crew turnover and reduced availability of training instruments to maintain perishable skills. The key for deployed commanders is to maintain a readiness level through the "out chop" point that is sufficient to counter the most likely threat scenario. The complexity of assigning units of measurement to the vertical scale is appreciated when one considers the quantity of factors involved with a 12-ship/10-squadron airwing/10,000-man Battle Group. By its very nature, Training Readiness is the hardest of all readiness components to quantify and therefore the most vulnerable to arbitrary funding cuts.

The difficulty in developing an acceptable method of linking training to readiness hampers our ability to quantify our training readiness posture and limits our ability to develop a feedback system to our training organization as depicted in Figure 2. Without proper/accurate feedback, the training organizational system is hampered in the decision making process. In an attempt to identify readiness indicators for feedback analysis, we in the Training Policy Division have identified over 180 existing potential data sources. The predominant source for

the various "measurement systems" have been taken from Type Commander (TYCOM) Training and Readiness Manuals and from the personal experience of officers from the various warfare specialties. These data sources may have value in assessing training and its contribution toward readiness.

Recent CNO initiatives, such as Total Quality Leadership (TQL) and the Navy Training Appraisal (NTA), have increased awareness in the importance of quantifiable data sources. Such high level interest has lead to the refinement in some of the 180 data sources. Over the past three years, the results have not only created more accurate measurements but, in turn, more useful ones. Continued efforts will certainly provide refinement in, both, data and data collection efforts.

CONCLUSIONS

Previous attempts toward establishing viable Training Measures of Effectiveness have represented independent, disjointed, and uncoordinated efforts. The need for a coordinated assessment tool is essential to the successful development of training to readiness links. To this end, I propose the development of a model which is based on the collation of required training support services as determined by FLTCINCs efforts to refine tactical training requirements. These support services should consist of items such as exercises, steaming/flying hours, and utilization of training ranges, targets, and ordnance. From these support services provided, a correlation can be formulated which relates to the unit and battle group exercise/inspection performance identified in the 180 data sources mentioned earlier. Thusly, a measured level of performance/readiness could be determined. The model could be used to assign specific readiness levels to the vertical axis of the readiness curve depicted in Figure 6.

The validity of this proposal hinges on the accuracy of the evaluation techniques used in determining the 180 measuring devices. The importance of this cannot be over emphasized. Detailed performance criteria must be strictly followed to allow assignment of meaningful grades so that variations in performance levels can be isolated to tangible process changes. This, in effect, creates a feedback mechanism called for in Figure 2 and one that is based on modeled information. A follow-on comparison analysis, based on properly defined and mutually agreed upon criteria, of the various performance indicators may provide an inroad to our goal of training resource optimization.

ACKNOWLEDGEMENTS

I would like to thank Mr. John McCloud and his staff of the PPBS Oversight and Execution Branch, Total Force Training and Education Policy Division, Office of the Chief of Naval Operations for the research and development of the estimates used to depict Individual and Fleet Training Costs. Additional thanks are extended to CAPT Bob Hechtman, Director Total Force Training and Education Policy Division, for his guidance, insight, and recommendations and LCDR Greg McCannel for his support in the development of the graphics portions of this paper.

ABOUT THE AUTHOR

CDR Ray graduated from the United States Naval Academy in 1975 and the Naval War College in 1987. He holds a Master's Degree from Salve Regina College and has been on the staff of the Director, Total Force Training and Education Policy Division (OP-11) from April 1989 to present.

"EMBRACING THE DEMONS OF TRAINING DEVICE ACCEPTANCE TESTING - THE PROCESS IMPROVEMENT LEGACY"

F.J. Winter, Jr., Director of Acquisition Support
Training Systems System Program Office, Aeronautical Systems Division

ABSTRACT

Under the auspices of Total Quality Management, a small group of Government and industry specialists examined the existing training device acceptance test process for potential improvements. The agreed-to mission of this Air Force/Industry partnership was to identify and promote implementable approaches to minimize the cost and time required for acceptance testing while ensuring that validated performance supports the operational training requirements. Application of a process improvement model focused on the customers and their requirements, analyzed how work was accomplished, and led to the identification and elimination of several non-value added components in current test practices.

Diverse technical and management approaches were blended into a single improved process known as Simulator Test 2000 (ST 2000). ST 2000 integrates timely, accurate, streamlined test documentation, provides safeguards for increased confidence in contractor verification testing, and improves on-time test milestone performance via an optimum balance of government/contractor specification performance validation procedures. By testing at a functional level in lieu of detailed testing constructs, this customer oriented approach emphasizes operational checks to determine ability to satisfy training objectives and eliminates Government repetition of previously conducted contractor tests. ST 2000 methodologies have been melded into both new and ongoing Air Force training initiatives. Further improvement highlights are those for contractor test performance incentives and commercial-type warranties.

To significantly reduce the number of Government test requirements, the joint Air Force/Industry team has formulated a total of 27 complimentary recommendations surrounding the test process. These improvements are estimated to save in excess of 40 percent of Government test time without compromising test objectives. This paper describes the development of these training device acceptance test improvements and the status/results of their implementation.

INTRODUCTION

When late during the summer of 1989, President Bush selected John Betti for nomination as Under Secretary of Defense for Acquisition, Total Quality Management (TQM) was fast becoming more than a household word. In fact, TQM was on a course destined to become an intrinsic management philosophy within the Department of Defense (DoD). For purposes of this paper, consider TQM as a leadership philosophy that creates a working environment which promotes trust, teamwork, and the quest for continuous improvement. Other essential elements of TQM require dedication, conviction, and a willingness to bring about change, to do the right things right, the first time, with the ultimate goal being customer satisfaction.

At about the same time as Betti's nomination, the Training Systems System Program Office at Aeronautical Systems Division (ASD/YW), Wright-Patterson AFB, Ohio was plowing new and fertile ground with contractors from the training system industry. Chartered in August 1989, the YW/Industry Total Quality Steering Group developed a mission "...dedicated to continuous process improvement and the acquisition of training products and services to produce the best trained aircrews and maintenance personnel in the world". The primary thrust of this Government/industry forum was to identify and provide recommendations to improve high level cross organizational processes having a critical impact on satisfying the customer's requirement using the principles of TQM.

THE CRITICAL PROCESS TEAM. The first Critical Process Team (CPT) chartered by the Steering Committee was to investigate aircrew training device acceptance testing. The team was tasked to thoroughly study the acceptance test process and recommend actions to improve test methodology. Membership represented a crosssection from the training systems development industry and included the following companies:

- o CAE-Link Corp
- o ECC International Corp
- o FlightSafety Services Corp
- o Hughes Simulation Systems Inc
- o Loral Defense Systems Division
- o McDonnell Douglas Training Systems Inc

Membership from the SPO consisted of two functional experts representing the disciplines of Engineering and Test Management.

The Cumberland Group, a subsidiary of Armco Steel, conducted an intensive four day workshop to train CPT members to analyze and improve the process. The Training Systems SPO agreed to fund the training for each CPT member. Team members gained a common understanding of the CPT purpose and were able to come to a consensus on how the acceptance test process is a summation of activities which must be completed in the course of providing a product or service. Effective working relationships were established and the team structure was created. Training provided the beginnings of an un-

derstanding of the Cumberland Process Improvement Model methodology.

THE PROCESS IMPROVEMENT MODEL. The objective of process management is to focus on the customers, determine what their requirements are, analyze how work will be accomplished, and identify and eliminate the sources of waste in the process.

The Cumberland Process Improvement Model consists of five primary steps leading to the elimination of nonvalue added components. The model stresses that quality problems are very often rooted in the process that produced them. A change in the process, therefore, is required to achieve meaningful improvement and not just merely eliminate the symptoms. This is the foundation upon which the process management approach to quality improvement is built. Following is a brief description of each step in the process improvement model.

Definition of the Improvement Opportunity: To develop a clear understanding of the team's task, desired expectations were clarified. A flow chart of all acceptance test activities was constructed to better understand the process. Finally, indicators (measures) of improvement were agreed upon to guide the team as it searched for areas of process adjustments.

Data Collection: A questionnaire, based on the measures of improvement was developed and used for data collection to move from the statement of the problem to a more complete description of the current process. Benchmarking of similar processes was initiated. Performance measures of several completed Government test programs were then documented for subsequent process analysis.

Analysis of Improvement Opportunities: Data collected from the previous step was used to identify and prioritize waste and focus efforts on the high payback areas. Waste is defined as any activity that does not add value to the process and was viewed as the primary opportunity for improving the process. In the final element of this step, the root causes of each major waste area were identified.

Development of Solutions: The intent here was to generate alternative ways to eliminate root causes of waste. The team concentrated on ways to significantly change the process instead of merely making minor adjustments. With no assumed constraints, the "perfect" process was visualized to form an understanding of what could really be achieved, even if in stages rather than all at once.

Improvement Recommendations: This final step was designed to improve the current acceptance test process through a series of recommendations resulting from solutions developed during prior analysis steps. Continual improvement is made by planning the modification, engaging the plan, then checking and making adjustments based on the results.

DEFINITION OF THE IMPROVEMENT OPPORTUNITY

The process of developing a clear understanding of the CPT's task and clarifying expectations for improvements in simulator testing was the team's first major assignment. The mission statement that follows was developed in order to clearly define the purpose and reason for existence of the CPT:

"We are the YW/Industry Partnership CPT, committed to continuously identify and promote implementable approaches to minimize the cost and time required for acceptance testing of aircrew and maintenance training devices and to insure that these devices support the users' needs."

The test process was initially defined as the primary means by which a training device is evaluated for compliance of the design/product against required characteristics and system performance. Through verification, validation, and authentication, the adequacy of performance characteristics are determined along with identification of deficiencies in system performance. Acceptance testing is defined as any and all contractor and Government activities performed to verify device conformance to specified system subsystem performance requirements.

The test process provides contract closure, and allows training initialization. Yet, despite its importance, the test process and accompanying test documentation has been reported as byzantine at best. Many myths and misinformation abound. There is widespread belief, for example, in the following:

- o Acceptance testing contributes to schedule delay
- o The Government must witness all acceptance tests

The flowchart shown in figure 1 depicts typical test activities and may vary somewhat depending on a particular program's requirements. Review of the existing process revealed that there were several test repetitions, multiple Test Readiness Reviews (TRRs), and numerous possible delay paths.

Process indicators were used to measure the performance of the acceptance test activities. Two approaches were used to generate the process indicators. First, the CPT membership produced a list of parameters which measured the performance of the acceptance test effort based on specific testing experiences. The second approach was to define measures relating directly to the delay loops in the process flowchart. Finally, the two lists were evaluated with reference to the following criteria:

a. If the test process is revised, will the proposed indicators be able to measure the improvement?

b. Can the indicator be measured in real terms using objective results?

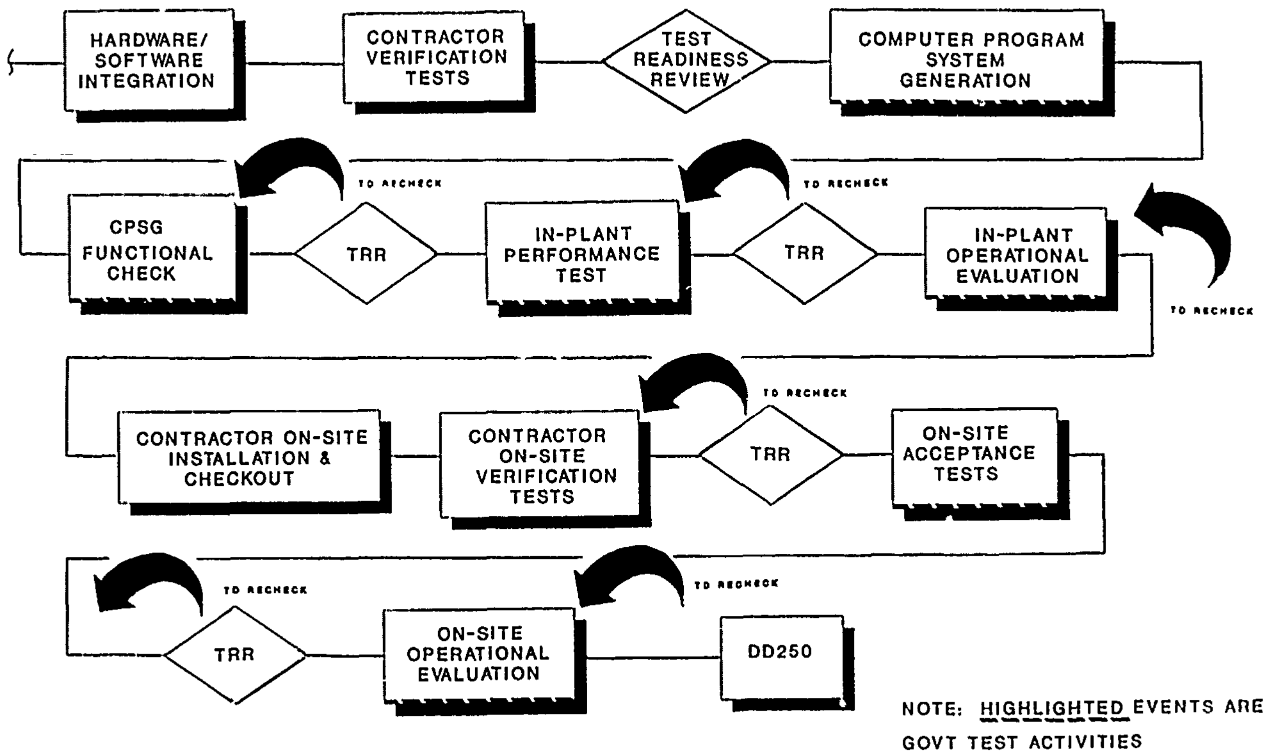


FIGURE 1. CURRENT SIMULATOR TEST APPROACH

c. Can data be obtained from the companies, is it something likely to be measured and retained as part of the existing testing process?

d. What is the most important data to request? The final list of process indicators totaled 38 distinct measures. The CPT focused on the four top indicators to collect data for further analysis.

Test Milestones Met or Delayed: The first indicator was based on schedule milestones required to conduct a test program. Program and contract schedule events were chosen in anticipation that such data would be recorded and available.

Number of Test Discrepancies: The number of Test Discrepancies (TDs) generated during acceptance testing is a measure of the training device quality. To get more insight into the causes of TDs, data was requested to include total number of TDs, number of TD re-submits, number of post sell-off TDs, and number of TDs out-of-scope.

Number of Days in Test: The purpose of this indicator was to isolate schedule variances by measuring duration of key test events (planned days vs. actual days). From the results, the CPT selected three phases to measure test duration as a performance indicator. These were in-plant development tests and on-site acceptance and operational evaluations.

Test Documentation: The CPT membership considered test documentation excessive. The size of the test procedures, i.e., number of pages, was the means used to measure this excess. In addition, the detail to which the procedures were written was measured by the number of test steps per page.

DATA COLLECTION

After settling on which indicators to be used, it was then necessary to consider the possible sources of data for the information the CPT needed. In determining the selected programs, the CPT focused on recently completed test programs and the likelihood of gathering accurate data. Finally a questionnaire which focused specifically on the process indicators was developed and used to gather supporting data.

The raw data from the questionnaire was analyzed and incorporated into summary sheets. Several very important adjustments were made to produce the summary sheets. The first adjustment was to eliminate companies/programs that did not respond to the questionnaire and those who felt their process was different. In addition, programs were excluded if sufficient data was not available. For example, only limited data was provided for maintenance trainer programs apparently due to less stringent test requirements. Comparison data was therefore nonexistent. As a result, six military programs remained for further evaluation.

BENCHMARKING. The concept of "who does it best" is known as benchmarking. The approach was to identify possible candidates for the CPT to evaluate as "best." The benchmark selection criteria included stringent testing and the use of commercial practices relating to the CPT process indicators. Candidate sources were:

- o Airline simulator programs
- o Simulation industry
- o TQM award winners
- o Other TQM intense companies
- o NASA simulators

The replies to the CPT membership inquiries and questionnaire were extremely poor. Successive follow-ups by team members did little to elicit further responses. Many indicated they felt their testing approach was sufficiently different to make it unsuitable for our purposes. It rapidly became apparent that integration and test of a full flight simulator is a uniquely challenging task not commonly encountered in other industries.

At that point the team decided to focus on the commercial airline simulator industry as the candidate for "best". This was based on the fact that they buy/build a product similar to the Air Force, use commercial standards, and must pass stringent acceptance testing conducted for/by the FAA. Seven commercial devices provided data considered adequate for benchmarking.

ANALYSIS OF IMPROVEMENT OPPORTUNITIES

There is a fine distinction between a problem and an opportunity. In this phase of the CPT effort, as problems were substantiated, opportunities became apparent. The predominant issue was then to focus/select opportunities that satisfied the mission statement.

After being reviewed for omissions, the raw data was organized for the purpose of identifying waste areas in the test process and subsequently determining the root causes. The data was grouped according to the four process indicators and studied for information and/or conclusions that could be drawn from the data sets. The data was plotted to obtain a visual representation and studied to identify relationships, trends and observations.

Each chart was individually reviewed to search for waste areas in the test process. Graphical analysis assisted in producing a better definition of the problems. The team used tabular data, histograms, the process flowchart, and comments on the questionnaires for identification of process waste areas.

For the reader with greater interest in the experimental design the complete data collection and analysis methods are contained in USAF TR-90-5000, 12 Dec 90, "Process Improvements in Training Device Acceptance Testing - A Study in Total Quality Management" available from Defense Technical Information Center (DTIC), Cameron Station, Alexandria VA 23661.

Eight critical waste areas were identified for subsequent root cause analysis. Root causes were uncovered by systematically questioning why the waste exists until the root cause is identified. This was done by asking "why" five times. This technique, while quite basic, allowed root causes to be determined for each waste area.

DEVELOPMENT OF SOLUTIONS. The solutions described are based on eleven months of intensive study, data collection, and analysis. With the knowledge obtained up to this point, the current test process flowchart was revisited. By removing all constraints, bias, and myths, then applying insight gained from the data, an idealized flowchart was generated to visualize a test process void of identified wastes. This new flowchart, in conjunction with the information gained from identifying root causes, provided the basis for developing solutions. Although the solutions are specific in nature, they are not intended to be perceived as the only solution but rather as the CPT's suggestions based on the research conducted. It should be noted that all solutions had a consensus of the CPT membership.

Waste Area 1: Delay in start of test.

The following root causes were identified by the CPT as being directly related to test delays:

- o Late Government identification of minimum training needs
- o Poorly defined requirements
- o Incomplete design
  - Requirements not complete
  - Data not available
  - Resources not available
  - Inefficient implementation of new technology
- o Manufacturing not complete
  - Government Furnished Equipment (GFE)/ Contractor Furnished Equipment (CFE) not available
  - Inadequate subcontractor/vendor management
- o Hardware/Software Integration in process measurement criteria lacking

Program delays due to late Government identification of training needs often cause the program planning phase of the procurement process to be incomplete. Inadequate research during this period results in poorly defined phenomenon during later stages of design development. The problem is further compounded when the contractor accepts these nebulous requirements and consequently fails to perform to Government expectations. Thorough completion of the training requirements analysis prior to the release of the RFP will greatly assist in well-defined, realistic requirements to provide a sound basis for contractor scheduling, pricing, and technical performance.

Problems associated with the contractor's failure to complete the design prior to testing due to incomplete data may be decreased or eliminated by identifying mission data early during the program and establishing joint

contractor/Government interpretation. This interpretation should then be formally included in the Design Criteria List (DCL). Early involvement of Sub ject Matter Experts (SMEs) will also help alleviate problems associated with a lack of data by providing an "on-line" data source during the design development phase. In addition, implementation of the data generation and management principles identified in the Simulator Data Integrity Program sponsored by the Training System SPO under contract AF33657-88-C-2168 should be considered to ensure accurate and complete data is provided by the weapons system prime contractor.

The problem of unavailable resources centers around delays caused by events leading up to and including Hardware/Software Integration which have been determined to be especially significant by this CPT. For example:

o Hardware/Software Integration suffers from poor planning and implementation.

o Ability to manage the Hardware/Software Integration process has been lacking.

o Start/stop criteria and in-process measurement tools have been non-existent.

Schedule risks associated with utilizing new technology in the device design can be mitigated by developing prototype testing procedures to mature the technology prior to Hardware/Software Integration. These procedures can be reduced on subsequent production quantities as the risk of the technology decreases.

GFE availability problems can be reduced by implementing a system in which the Air Force procures the required training system components from the prime weapons system contractor as soon as GFE requirements are known. An alternative is to have the Government include in the weapon systems contract the requirement to enter into an associate contractor agreement with the simulator contractor to supply the necessary components. Alternatives to using GFE should be explored such as the use of commercial components which are essentially equivalent to MIL-SPEC hardware items.

Inadequate subcontractor/vendor management problems are not in the customer's direct line of responsibility; however, the Government can influence the prime contractor to address this area to reduce the risk to testing. Suggestions by the CPT for industry improvements include:

o Avoid multi-level, multi-party subcon- tractor arrangements on major device components where the actual supplier has no direct link to the prime

o Establish a strong subcontractor/ vendor management team to take responsibility for the supplier's performance

o Use on-site representatives when necessary to closely monitor supplier performance

o Use Material Requirements Planning (MRP) packages to help schedule vendor delivery

o Develop reliable second sources for high risk components

o Insist on monitoring and reviewing major subcontractor performance on a regularly scheduled basis to identify potential problem areas

Waste Area 2: Redundant testing.

The following root causes were identified as contributing to the problem of redundant testing:

- o No Government recourse after buy-off
- o Improper engineering test procedures
  - Engineering procedures not repeatable
  - Results not documented

The customer typically views acceptance testing as his "one and only shot" at discovering all system problems. This results in aggressive testing by the Government to ensure the continued performance of the simulator throughout the required life cycle. The contractors can instill confidence in their product and thus lessen the need for extensive testing by providing a more comprehensive performance warranty package similar in scope to those currently available to commercial airlines.

Redundant testing often occurs as a result of poor contractor testing procedures. The Government does not accept Contractor Verification Testing (CVT) results as "final" and usually reruns a substantial portion of contractor test procedure. Confidence in contractor test results can be established by increasing the quality of in-plant testing and including advisory SME involvement during CVT. This solution prompts the contractor to conduct in-plant tests which are repeatable and well-documented. Consistent contractor test results will increase the likelihood of Government acceptance of the data generated and eliminate the need for repeating previous contractor testing. Failure to properly perform and document CVT also results in jeopardized on-site device performance and reduced profit due to attending schedule delays and additional contractor resources necessary to upgrade the device to an acceptable condition.

Waste Area 3: Detailed customer subsystem Performance verification.

The CPT identified the following root causes as contributing to overly detailed performance verification testing:

- o Contractor test results not available or documented
- o Traditional, bottoms-up test techniques
- o Performance risks associated with new technology

The need for detailed Government performance verification to the subsystem level can be eliminated by instituting improved contractor test procedures. Prototype tests should be developed for high risk, new technologies prior to Hardware/ Software Integration until a satisfactory confidence level is reached. Traditional bottoms-up testing should no longer be performed by the Government. Instead, these procedures should be completed during

contractor testing. The procedures and test results should be thoroughly documented for Government review, thus allowing one time cost effective and efficient system level acceptance testing.

Waste Area 4: Test Discrepancies.
Examination of TDs as waste areas yielded the following root causes:

- o Lack of trained resources
- o Invalid test procedures
- o Poorly defined operational performance
- o Data shortfalls
- o Incomplete contractor testing

Proper training of test personnel in these areas is essential in limiting the number of discrepancies written against a given device. Proper training ensures that test procedures are properly set-up and performed and test personnel are able to sufficiently measure device performance against performance criteria. Implementing in-house training programs based on the principles of TQM to create a climate of pride, teamwork, and ownership has great potential to alleviate excessive write-ups and increase overall quality.

Poorly defined performance requirements also contribute to unnecessary TDs. This problem relates back to a failure of the Government to completely identify training needs early in the program.

A closely related problem is a lack of performance data. Data shortages can be alleviated by making SMEs available throughout the design, development, and contractor testing phases of the program. SME involvement in design reviews is especially encouraged to clarify design data assumptions and resolve ambiguities with the results then formally documented.

It should be noted that TDs are symptoms, not causes. Root causes which give rise to TDs have been identified and solutions discussed in several of the other waste areas.

Waste Area 5: Excessive test Documentation.
Contributing to the problem of excessive documentation, the following root causes were identified:

- o Documentation is overly complex and detailed providing for
  - Repeatability
  - Skill level
  - Support considerations
  - Test Matrix requirements
- o Documentation is not coordinated across contract requirements
  - Micro management

Overly complex and detailed documentation can be alleviated by writing engineering tests at the functional level. Greater emphasis should be placed in the requirements for contractor development of automated test procedures for such areas as aero and performance tests, initial conditions, avionics, nav-aids, diagnostics, etc. A further solution is to have Government testing at the mission level which eliminates the need for step by step procedures. Another solution is to examine the Test Matrix at design reviews to minimize the requirements for tests and demonstrations based on the evolving systems design.

Waste Area 6: Test Interruptions.
The following root causes were identified as causing schedule interruptions:

- o GFE/CFE spares not available
- o Schedule pressure
  - Acceptable risk
- o Poor systems analysis/solutions unsatisfactory
- o Customer facility not ready
  - Lack of control of the construction program
  - Lack of contractor design
- o Software update process errors

Test interruptions due to schedule pressure often result from allowing known problems to exist unresolved. The contractor should monitor these problems through the use of established risk management procedures, resolve TDs as quickly as possible, and ensure trained personnel are available in each specific program area. The determination of acceptable risk to enter into Government testing with known problems occurs during TRR. A more comprehensive analysis effort should be made at this time prior to entering test to avoid delays, disruptions, and waste.

A lack of GFE/CFE spares is often responsible for test interruptions. The Government can solve this problem by considering spares requirements when ordering components from the weapons system manufacturer.

Poor system analysis is the root cause of many test interruptions as unresolved critical TDs often result. An ad hoc team should be established to resolve "show stopper" TDs utilizing both contractor and Air Force resources as required.

Modification or new construction of a facility is most often accomplished via the Military Construction route (3300 appropriation funding). However, the ability to use RDT&E (3600 appropriation funding) should be exploited where new facility construction is a requirement. AFR 80-22 states that RDTT&E funds may be used to acquire industrial and RDT&E facilities needed by contractors to fulfill R&D contracts as authorized by 10 USC 2353. This has been interpreted to mean that where a facility is needed by a contractor in order to perform tasks required by a R&D contract, that facility may be provided through this funding.

Waste Area 7: Multiple test readiness reviews.
A singular root cause was identified generating multiple TRRs:

- o Failure of the contractor to be ready for test

Multiple TRRs are costly to both the Government and contractor, contribute to the length of the test schedule, and constitute a non-productive expenditure of test team resources. Data indicates that the contractor historically is not ready at TRR. Multiple Government TRRs can be avoided by placing the burden of test readiness solely on the contractor. The need for multiple TRRs should be re-evaluated and contract requirements written to reduce the number of TRRs accordingly.

Waste Area 8: Computer Program System Generation (CPSG).

The following root causes were identified as contributing to CPSG requirements:

- o Lack/weak contractor software tools
- o The requirement to accommodate changes

CPSG requirements constitute a lengthy and unnecessary waste of resources when large scale, complex devices are involved. Software tools and processes are available which provide the same level of confidence in the integrity of the software as CPSG. This is particularly true for those programs requiring Ada software language or otherwise require automated configuration management routines. Government certification of contractor software tool and processes for software configuration management and the capability to support changes should be accomplished prior to test. Certification would then allow for the elimination of CPSG requirements.

IMPROVEMENT RECOMMENDATIONS

The CPT is recommending fundamental changes to the acceptance test process and activities supporting that process. These recommendations are based on analysis of the current test methodology and solutions formulated by the team to eliminate the root causes for identified wastes. The recommended new process known as Simulator Test 2000 (ST 2000) is shown in Fig 2. When contrasted against the current test process (ref Fig 1), the elimination of redundant Government CPSG testing, in-plant performance tests, and on-site acceptance tests becomes obvious. Accountability will be improved by better aligning authority with responsibility for Government and contractor development test teams. Unnecessary testing will be eliminated, test documentation will be minimized, and the level and type of testing will be more focused on satisfying training requirements. In addition, a comprehensive and more effective test assessment will be realized without extensive TRRs. Test procedures will be elevated to the functional level and Air Force Subject Matter Experts (SMEs) will be made available to assist the contractor during systems development. Test functionality will not be reduced nor will test integrity be compromised as a result of the recommendations proposed in ST 2000.

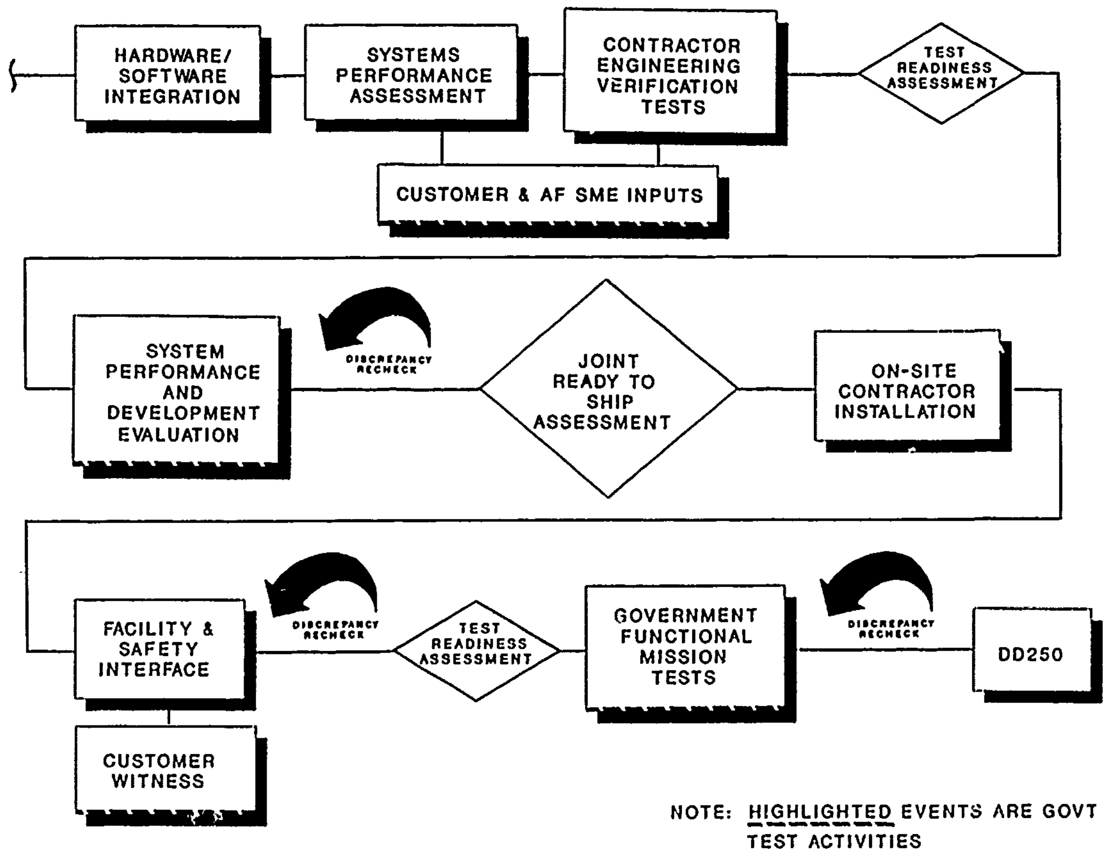


FIGURE 2. SIMULATOR TEST 2000

The CPT concentrated on the task of improving the efficiency of acceptance testing. However, "testing" encompasses and is influenced by a much broader span of activity outside the formal test program. Program development tasks prior to acceptance testing were suspected by the CPT membership of masking problems which subsequently appear during or delay the start of acceptance testing. The data collected supports prior suspicions.

CPT findings show a major cause for delay in fielding acceptable training devices is due to activity that precedes the start of acceptance testing. In particular, recommendations made in the areas of design data, aircraft components, and Hardware/Software Integration are emphasized because of their known historical impact on the test program. Correction of these problems will largely avoid significant delays experienced on past programs.
With regards to design data, the CPT recommended that SMEs be made available to the training device contractor early in the program to help resolve data deficiency problems and establish consensus on interpretation and application. Further, that the Government implement the Simulator Data Integrity Program study recommendations to ensure timely, accurate, and complete data availability to the training device developer from the weapons system design contractor. Also recommended is the use of more aggressive, comprehensive performance warranties to crystalize contractor liability and bolster Government confidence in contractor assertions to "meet the specification." This is hoped to radically reduce contract test requirements. Lastly, a well defined and realistic Training Systems Requirements Analysis should be developed prior to release of the RFP to establish a bound training tasks early in program development.

To relieve the schedule delays attributable to unavailable or late aircraft components, early identification of requirements (including spares) followed by obtaining sufficient priority for timely acquisition from the weapons system contractor is considered essential. Components could be manufactured or alternately provided by the training system developer via an associate contract agreement with the weapons system developer. Prime contractors are encouraged to strengthen subcontractor/vendor management processes to improve delivery performance and reduce the impact on device readiness. A repair pipeline, negotiated with the original equipment manufacturer prior to testing, may insure the availability of spares during the test program.

Significant recommendations to improve hardware/software integration include establishing an Industry lead CPT to investigate viable HSI in-process measurement criteria. Improved management of this important development step is critical to Contractor Engineering Verification Tests (CEVT). To mitigate technical performance and schedule risks use of prototype testing to mature new technology applications prior to attempted insertion into Hardware/Software Integration is proposed. To abate the impact of test interruptions, risk management programs must be developed and implemented to anticipate and manage possible causes. Training device development programs requiring new/modified facilities should consider tasking the development contractor as authorized by 10 USC 2353 to centralize contract engineering responsibilities to insure facility availability.

The CPT recognized that the contractor currently has every incentive to start Government test to see if he can "selloff" the device and save schedule. If testing fails to achieve the desired result, the contractor may find it more economical to resist corrections, attempt short term solutions, and hope test schedule concerns will cause the Government to weaken its position.

However, the CPT also believes that exceptional contractor test performance should be rewarded. Conduct of an effective, well planned test program is a worthwhile objective. The creation of contract incentives to accomplish this, however, is dependent upon several variables including the basic nature of the testing (development vs. production), the type of contract (cost vs. fixed price), and the type of incentive (i.e., objective performance incentive versus subjective award fee incentive). These factors, along with other pertinent facts, must be weighed when trying to assess the ability to create a "real" contract motivator.

Development Testing - Cost Type Contract - Award Fee: Because of the very nature of a development program, it would be extremely difficult to structure a performance incentive which would be meaningful. Development testing by its very nature is intended to surface problems before the design is frozen and moved into production. The key is solid test planning and analysis in order to minimize surprises. These areas tend to be quite subjective when attempting to establish measurement criteria. Award fee provisions would allow tailoring of the incentive from period to period as the program progresses and provide multiple opportunities to reward performance.

Production Acceptance Testing - Fixed Price Type Contract - Performance Incentive: In a production environment, test requirements are usually quantifiable. Because of this, the Government's ability to write a meaningful incentive at the time of contract award is much greater. A concern remains that the incentive being structured is sufficient in terms of dollars or corporate visibility to be an effective motivator and is in balance with the remainder of the program. From a cash flow or liquidation standpoint, the contractor is still motivated to push for contract buy off in order to claim progress payments and profit. Additionally, the incentive must be justifiable in terms of overall savings to the Government (i.e., reduced Temporary Duty, more efficient use of personnel, etc.).

The CPT recommended that initiatives begun by The Training Systems SPO on the Simulator for Electronic Training (SECT) and Special Operations Forces Aircrew Training Systems Programs (award fee) and the Digital Radar Land Mass System (performance incentive) under development for the F-16C/D weapon system trainer be

continued and applied to all new acquisition programs. The need to monitor results of programs where incentives have been applied is considered extremely important to ensure the level of value is worthwhile to the contractor, that the incentives remain realistic and achievable, and experience gained through their application is reinvested in future programs.

CONCLUSIONS

The premier improvement to simulator and maintenance trainer test process has been identified as Simulator Test 2000. Reforms to reduce Government test, strengthen and reallocate contractor test responsibilities, refine test documentation, and discrepancy management encompasses major CPT recommendations.

For any recommended process change to be considered for implementation, a measurable improvement must be expected. If there is a significant anticipated benefit as a result of the new process, a high degree of motivation to adopt the new process will be present.

A comparison of the current simulator test approach shown in Fig 1 and referred to here as the "Idealized Weapon System Trainer (WST) Test Program", can be made to the ST 2000 process. The Idealized WST Test Program assumes that once testing begins, it progresses and is completed without delays or interruptions. This test program includes CVT, Performance and Acceptance Tests, and in-plant as well as on-site Operational Evaluations as depicted in Table 1. The total test effort of forty-eight (48) weeks required by the idealized WST test program can then be compared to estimates for a WST using the ST 2000 process as shown in Table 2. It can be seen that the reduction of the total test effort from forty-eight (48) weeks to thirty and one half (30.5) weeks represents a savings of approximately thirty-seven (37) percent.

The consensus of the CPT members is that with the adoption and implementation of the ST 2000 process, a significant savings in resources can be realistically expected if all elements of ST 2000 process are implemented.

Additionally, if the other recommendations and suggested changes are also implemented, increased efficiency in training device acquisition programs can be achieved. These are outside the formal test phase, but directly affect the start or progress of testing. The potential savings to cost/schedule which can be achieved by implementing these changes are not to be overlooked. The largest waste in most military training device development programs occurs, for a variety of reasons, prior to the device being ready for testing. Data collected showed that on average military programs are 132 days (26.5 work weeks) late prior to beginning of test. The elimination of this waste would result in cost and schedule overrun savings of approximately eighteen (18) percent over the life of a planned thirty-six (36) month program.

Table 1. Idealized WST Test Program

		IN-PLANT		ON-SITE		
TESTS:	CVT	Performance Test	Ops Eval	Acceptance Test	Ops Eval	TOTAL
WEEKS:	20	18	2	6	2	48

Table 2. ST 2000 Process

		IN-PLANT		ON-SITE	
TESTS:	System Assessment	CEVT	SPADE*	Functional & Mission Tests	TOTAL
WEEKS:	1.5	20	6	3	30.5

* SPADE = System Performance and Development Evaluation

The Simulator Test 2000 concept has been embraced by the Training Systems SPO and implemented on new program starts. The Simulator for Electronic Combat Training Development Program, the Leading Air Training Command Electronic Warfare Officer Training Initiative, and the Euro-Nato Procedures Trainer Modernization Program are both structured to take full advantage of the ST 2000 process.

The development requirements for aircrew training devices for the Joint Surveillance Target Attack Radar System will similarly adopt the ST 2000 improvements and an on-going contract for C-17 maintenance training devices is being modified to take advantage of test cycle process improvements and reduce government test program requirements (man-months) by an estimated 50 percent.

The most significant process change is the customer's agreement to accept CEVT test results. Repetition of these types of specification compliance tests by the Government are no longer a requirement. This commitment eliminates the single largest cycle of customer testing from the current acceptance test process.

In point of fact, ST 2000 places the responsibility to thoroughly execute CEVT squarely on the contractor. It must be conducted to the same level as required for developmental performance testing previously conducted by the Government. This means that test results must be documented, verifiable and repeatable. Failure to execute stringent, valid testing with documented results will motivate the customer to demand a repeat of previously run tests and will again require 100% witnessing of CEVT. If the contractors do not perform their part, the customer has no alternative but to revert back to the existing test philosophy. The customer has extended the opportunity, the contractor must aggressively respond for ST 2000 to be viable.

ENDNOTES

1/ Winter, F.J., et. al, "Process Improvements In Training Device Acceptance Testing - A Study In Total Quality Management" USAF ASD TR-90-5000, Dec 12, 1990

2/ Shaw, J. and Lloyd, William "Bridging the Information Gap." Proceedings of the Eleventh Interservice/Industry Training Systems Conference, (November 1989)

ACKNOWLEDGEMENT

The author wishes to acknowledge the extraordinary participation of the following professionals as members of the CPT:

-Bill Adams
Hughes Simulation Systems Inc
-Al Emerson
ECC International Corp
-Jerry Kosydar
CAE-Link Flight Simulation Division
-Bob Lillie
Training Systems System Program Office, USAF
-Sam Malene
Loral Defense Systems Division
-Larry Stewart
CAE-Link Flight Simulation Division
-John Thorn
McDonnel Douglas Training Systems Inc
-Floyd Weaver
FlightSafety Services Corp

That the development of training device acceptance test improvements have proceeded so rapidly, to such successful outcome with such a far-reaching impact, is a testimonial to the vision, skills, and dedication of this highly motivated and professional team. Those useful insights contributed greatly to the development of this paper.

INTEGRATED AIRCREW TRAINING MANAGEMENT INFORMATION SYSTEMS:
AN ORGANIZATIONAL PERSPECTIVE

Robert T. Nullmeyer
Philip D. Bruce

Armstrong Laboratory
Aircrew Training Research Division

and

Marty R. Rockway

University of Dayton Research Institute

ABSTRACT

The trend in the United States military services is to design training programs as total systems rather than as collections of courses or blocks of instruction. This trend has highlighted the need to design an integrated aircrew training management information system (ATMIS) to ensure the cost-effective operation, maintenance, and evaluation of the total system throughout its life cycle. For the past several years, the Aircrew Training Research Division of the U.S. Air Force Armstrong Laboratory has been engaged in a field research program to identify the functional characteristics and information/data requirements of ATMISs. A number of military and contractor aircrew training systems have been reviewed and analyzed. The purpose of this paper is to discuss some of the findings and to propose a systematic approach for the design of ATMISs, with particular emphasis on the identification of comprehensive, multi-user information requirements. This approach is presented in the context of a new, contractor-designed and supported aircrew training system, which is intended to replace an existing Air Force system. The composition and use of representative multi-user working groups, a baseline analysis of the existing ATMIS, and procedures for determining the information requirements posed by the new system are discussed. These information requirements are developed from an organizational perspective. It is suggested that the entire sequence of ATMIS design, development, and operation be subjected to a rigorous test and evaluation process, including an assessment of its impacts on organizational performance.

INTRODUCTION

The current trend within the Air Force is to design aircrew training programs as <u>total integrated systems</u> rather than as collections of courses or blocks of instruction. This trend has been coupled with a concurrent shift to contracting the design, delivery, and support of aircrew training. These changes have introduced a new set of technical and management issues which impact the design, development, evaluation, and

operation of aircrew training programs. For the past several years the Aircrew Training Research Division, Armstrong Laboratory (AL/HRA) has been conducting research and development (R&D) to address several of these issues in order to provide principles, procedures, and user-oriented guidelines to support Air Force acquisition and operational training agencies.

During the conduct of its R&D efforts, AL/HRA has intensively reviewed a number of aircrew training programs, both Air Force and contractor supported. It has also been an active participant in a number of other programs, which involved substantial collaboration and interaction with Air Force and contractor personnel throughout the various stages of the aircrew training system (ATS) life cycle [1, 2, 3]. Among the programs with which HRA has been directly involved are the C-130 ATS with the Military Airlift Command (MAC) and CAE-Link; B-52/KC-135 Combat Crew Training School (CCTS) modernization with the Strategic Air Command (SAC); and the Special Operations Forces (SOF) ATS with MAC, the Air Force Special Operations Command, Aeronautical Systems Division (ASD) of the Air Force Systems Command, and Loral Defense Systems.

A key characteristic of all the ATSs considered to date is that they require an explicitly defined aircrew training management information system (ATMIS), usually with some degree of automation. In the case of the more complex ATMISs, the capabilities specified extend well beyond the traditional training-program scheduling and record keeping functions. For example, Dukes, Rockway, and Nullmeyer [4] described the training management system developed for the C-130 ATS. This system consists of eight modules: 1) administrative management, 2) resource management, 3) curriculum management, 4) scheduling management, 5) student performance measurement, 6) reports, 7) configuration management (including courseware), and 8) logistics management. A single, integrated database supports each of these functions. Another example is provided by Reakes [5], who has described the functional requirements associated with the training management system for the C-141 ATS.

From our research and experience with the various ATSs, it has become obvious that a properly designed, implemented, and utilized ATMIS is essential for the cost-effective operation and management of complex aircrew training programs. In addition, the enhanced capability to collect, process, and manipulate information in a variety of new and different ways can provide a means of gaining greater insights about many critical ATS issues and relationships. This proved relatively intractable under the previous way of doing business, because the data to resolve these issues were either not available or inaccessible.

Despite our optimism concerning the potential value of well-designed ATMISs, it has been our observation that most of these systems have fallen short of their promised capabilities. A principal reason for this shortfall is that the major focus of most ATMIS development efforts is on the definition of computer hardware/software and data processing capabilities, rather than the information needs of the organization responsible for operating and managing the training program. The purpose of this paper is to suggest some general ways, conceptual and procedural, of increasing the ability of ATMISs to benefit the larger aircrew training system. We shall try to accomplish this by 1) providing an organizational context in which to understand the design, use, and criteria for assessing the effectiveness of an ATMIS, 2) discussing the use of an integrated

information database to support multiple users, and 3) suggesting some procedures for determining the information requirements of the organizational elements, or users, of the ATMIS.

AN ORGANIZATIONAL PERSPECTIVE

In his book, Why Information Systems Fail, Lucas [6] has observed that in our concern over technology, we seem to have ignored the fact that information systems exist within the context of an organization. In other words, we have focused too much attention on the technology per se, instead of its uses within the organization. The culmination of this fascination for technology is the purchase of sophisticated computer systems for the management of information without first having completed the labor-intensive effort of understanding the organization and deciding precisely how the equipment should best be used to serve the organization. Development efforts can follow similar paths and occur in the relative absence of understanding the real-time organizational functions.

Our view is that we must first understand the training organization, its functions, the information it generates, and how information is used, in order to design and develop effective ATMISs. The individuals working in the aircrew training organization include instructors, evaluators, aircrews, schedulers, record keepers, curriculum designers, and a variety of managers at multiple levels. These functions, or jobs, are highly interdependent, and it is no surprise that the information network and flow which has evolved over time to serve this organization of people is also composed of interdependent elements.

An example, which is based upon an analysis conducted within the B-52/KC-135 aircrew training system [1], may be helpful for illustrating some of the interdependencies between organizational elements and the information system. The event requirements for qualification and continuation training programs are generated and coordinated from SAC Headquarters, and they are published in the 51-series training regulations (e.g., SAC Regulation 51-52). Flying hours are allocated to the units on the basis of these and other requirements. At the unit level, continuation training tables are constructed for each member of an aircrew, and they are stored in a computer which is part of the Air Force Operational Resource Management System (AFORMS). Paper products are often sent to the flying squadrons to keep squadron commanders, operations officers, flight commanders, and aircrews apprised of impending or completed requirements. Flying sorties must be planned, developed, and scheduled on the basis of these requirements so that aircrews remain on mission ready status, once they are qualified. A great amount of coordination between operations and (aircraft) maintenance is necessary to accomplish this objective, and its culmination is the publication of the weekly flying schedule, which is made available to all concerned parties, and flying the actual sorties by the aircrews.

At the conclusion of each flying sortie, aircrews must document training events accomplished, the amount of flying time consumed, and other items. The flying hours are entered into a separate database on the maintenance side of the wing. Training activity during the sortie must be verified by a process of mission review, and the flying events which are credited to the aircrews are fed back into the AFORMS computer--and the system is updated. The number and availability of mission ready aircrews are also reported daily in the Training Measured Area of the Status of Resources and Training System

report. The Chief of Aircrew Scheduling, who manages the wing flying hour program, tracks aggregated (all aircrews) training events accomplished. These events are then related to the overall amount of flying hours consumed, and they are monitored over the accounting period as the flying-hour budget is managed. Training events accomplished for the entire wing are also reported to SAC Headquarters in the Wing Commander's Training Review Panel Report.

While the implementation of a flying hour program is, of course, much more complex than depicted above, several points can be made from the illustration. 1) It is necessary to study in sufficient detail and understand exactly these kinds of dynamic organizational processes, which include all the information generated and how it is used, if we expect to be in a position to design effective ATMISs. 2) These studies must be conducted <u>prior</u> to designing new information systems or targeting improvements to existing systems. 3) Once the organization and information flow are redesigned, we should then select the equipment to implement that flow [7]. Finally, 4) the ultimate criterion for determining the effectiveness of a new ATMIS is that <u>organizational performance</u> is somehow improved. The ATMIS is envisaged as a means of accomplishing the improvements in organizational performance.

INTEGRATED AIRCREW TRAINING MANAGEMENT INFORMATION SYSTEMS

The aircrew training community has pursued integrated training systems in order to eliminate gaps and overlaps in its programs, and to improve overall effectiveness and efficiency. It has attempted to accomplish this by designing each component of the ATS in a way that best serves total system goals. Similar benefits should be attained by considering all the activities associated with generating, storing, processing, reporting, and using training information as a single, integrated subsystem of the ATS. This notion would apply whether the function is performed by Air Force and/or contractor personnel, and whether the information system is manual or automated. In using this approach, the goal is to satisfy the information needs of each relevant organizational element within the ATS and integrate them in the most effective and efficient manner possible. The genesis of this notion was our experience with ASD and MAC as we defined Air Force information requirements for inclusion in the C-130 ATS Statement of Work. MAC viewed high-quality training effectiveness data as a key to the successful transition from in-house military aircrew training to a joint Air Force and contractor system. In fact, a reference book developed by MAC for the C-130 ATS referred to test and evaluation as the single most important component of the ATS [8]. Access to high-quality data was also critical for the Laboratory to attain its follow-on research goals. As each Air Force participant identified their information needs, it quickly became apparent that there was a great deal of overlap in data requirements across functional areas. We took advantage of the commonalities in the data needed to support functions such as courseware validation, training system test and evaluation, product assurance, technical performance measurement, research, and quality control. The cost-effective solution was to design a single, integrated database which incorporated the information needs of each user and eliminated these redundancies. This was to become one of our central notions in the design of training management information systems.

We eventually learned, however, that specifying and consolidating the various training information

requirements is only one part of a much larger task of designing an integrated ATMIS. In order for the ATMIS to be responsive to the needs of the training organization (both Air Force and contractor elements), the designers must be able to specify which organizational elements are to be served by the system, the precise information needs of each user, and how each organizational function could be enhanced by an improved information system. While the need for such knowledge seems apparent, the required level of detail has proven difficult to attain in actual practice. The following section describes an approach for obtaining and coordinating the required ATMIS design information.

PROCEDURAL CONCEPTS AND PRINCIPLES

Formation of an Integrated Information System Working Group

All users must be active participants in the design process in order for the ATMIS to be responsive to each user's needs. Accordingly, we envision first establishing a fairly high-level user working group comprised of each major organizational element that would be affected by the new ATMIS (e.g., the acquisition agency, the Air Force site(s) where the ATS would be implemented, the MAJCOM Headquarters, and the contractor). The initial tasking for this group would be to identify the comprehensive information requirements of the joint Air Force and contractor ATS. The working group would focus on the training information needed in the operational phase of the ATS, but it would also address the earlier development phases. Each working group member would represent a much larger constituency and must ensure that constituency needs are met to the extent which is practicable. The activities of the principal working group would be augmented by the efforts of sub-working groups, including various experts, in order to determine these needs. This is because 1) it is unlikely that principal working group members would be knowledgeable about all the information requirements within their organization, and 2) considerable resources are required to accomplish an effort of the intended scope and level of detail. The eventual goal of the working group is to reach a consensus on the actual users of the ATMIS, the information needs of each user, how the collective information/data requirements are to be consolidated and provided for most efficiently, and the intended organizational improvements that would result from implementing the new system. It is anticipated that the resulting product would represent a coordinated Air Force and contractor position on ATMIS design and implementation. While this process is labor-intensive, it is necessary if the working group is to make well-informed decisions about the ATMIS. The remainder of this section describes the proposed process in more detail.

Baseline Analysis of the Existing Information System

The first major task in establishing requirements for the new ATMIS is to conduct a comprehensive and sufficiently detailed baseline analysis of the existing training management information system, including automated and manual components. The importance of this baseline analysis cannot be overemphasized. It is a necessary step to ensure that the newly designed ATMIS will actually serve the needs of the organization. The baseline analysis accomplishes at least three things: 1) it provides a common frame of reference for all parties to understand the current organization, its requirements for information, and how it uses information as organizational functions are executed; 2) it serves as the principal means of identifying potential high-payoff improvements or innovations; and 3) it serves as a benchmark for assessing whether

improvements in organizational effectiveness or efficiency actually result from implementing the new ATMIS. It is our observation that design and development activity which occurs in the absence of a sufficient baseline analysis of the existing system is destined for failure.

A few years ago, AL/HRA [9] conducted an analysis of the training information system which supports the B-52 and KC-135 CCTS at the 93 Bombardment Wing (BMW), Castle AFB, CA. The CCTS conducts initial qualification, pilot and navigator upgrade, and requalification training. Although the emphasis in the study was on the evaluation function, it was an attempt to document the information flow through that part of the organization which is responsible for the design, development/validation, delivery, management (at multiple levels), and evaluation of training. It was conducted from an organizational perspective, as discussed above, and it stressed how information was generated and used by the aircrew training organization.

The effort was initiated by studying the organization from a current organizational chart and being briefed by individuals knowledgeable about the 93 BMW aircrew training operations. This resulted in a general understanding of the principal organizational elements and how they functionally related to each other. It also enabled the construction of a working model which depicted the flow of training activity and how the organizational elements were involved in that process. The organizational elements included the academic and flying squadrons; the Standardization/ Evaluation Division; the Instructional Systems Development Division; Aircrew Training Devices; the Operations Systems Management Branch; Wing Scheduling; and the Deputy Commander for Operations, the Assistant for Training, and the Bomber and Tanker Training Management offices. In addition to the organizational structure and functions, major forums for the exchange of training information, training development activity, and decision making were identified, and it was depicted how these functioned within the organization.

With this basic model of the training organization as a point of departure, the organization was then envisaged as being "activated" or set in motion by a class of students entering for initial qualification training. The organization and its training information system were then studied as a dynamic, interactive process. The "hypothetical" time period which constituted this flow of activity extended from the in-processing of students at the CCTS, through the initial qualification training and certification process, to graduate assignments at operational units. The period of time was arbitrarily ended when graduates were in-unit for approximately six months and completed CCTS external evaluation questionnaires, which were returned to the 93 BMW. All crew positions for the B-52 and KC-135 were considered. Each organizational element was then visited to ascertain and understand: 1) what training or training related activity was performed; 2) what information was generated in the process; 3) what information was received from other parts of the training system; 4) how information was actually used in the execution of organizational functions; and 5) what information was then sent to which elements within and/or outside the immediate training system. In addition, meetings of the working groups and review boards/panels were attended, and their functioning was observed--including what information was presented, discussed, and acted upon.

It is necessary to work intensively with the people who

actually perform the training and training-related functions, in order to develop the required understanding of the organization and its information system. This is not a task that can be accomplished by sitting down with one person in a training wing and having them attempt to explain the entire system. This is partly because it is unlikely that any single individual in the organization is thoroughly knowledgeable about each of the functions, let alone all the information that is generated and how it is used. Accordingly, a baseline analysis involves considerable over-the-shoulder experience: observing and interacting with people as they perform their jobs, including how they generate and use information.

In addition to intensive interviews and participant observation, it is necessary to examine a considerable amount of archival data. In the study of the 93 BMW, these sources included student data forms for academic and flight-line instruction, charts, reports, and other products of record keeping systems, both automated and manual. A considerable amount of data is typically stored in file cabinets for a period of time set forth in a regulation. In addition, regulations, operating instructions, and other policies and procedures are important sources of information by which to understand the various organizational functions and the information gathering and reporting activity. The baseline analysis team then has the task of "piecing together" the information flow for the entire organization on the basis of all the possible sources at its disposal. The results must then be validated by individuals and groups within the organization.

Development of Requirements for the New ATMIS

The baseline analysis provides a strong foundation for the working group to develop information and automation requirements for the new ATMIS. The working group must first map the baseline training organization--which is Air Force only in this example--onto the new training organization, which is composed of Air Force and contractor elements. This provides a structure for accomplishing the next step, which is to translate the existing information requirements into information requirements for the new system. To the extent that organizational functions remain the same, we would expect the results of the baseline analysis to be directly applicable. If organizational functions are changed, added, or deleted, information requirements would have to be readdressed. A process similar to that followed in the baseline analysis would be used to develop the entirely new or modified information requirements: determine what organizational function is to be performed, how it is to be performed, and what information is necessary to execute the function. Individuals who are to perform these organizational functions must be consulted, as they are an important part of the requirements development process. As the new, modified, or intact organizational elements and the corresponding information requirements are sorted out in detail, the working group must also consider any proposed high-payoff innovations or improvements, including those identified during the baseline analysis. These will undoubtedly include recommendations for the automation of certain information collection, processing, and output functions; but they may also include better ways of performing organizational functions, such as evaluation of the ATS, which may also have information or hardware/software implications.

An important function of the working group is to consider various ATMIS design alternatives and resolve

trade-off issues which arise among the competing configurations. Automation costs money, but it must be clearly recognized that information (data) also has important cost implications. Different kinds of data, such as student critiques of instruction or valid knowledge and performance measures collected during the instructional process, have different costs and values associated with them [9]. As the working group converges on a final design of the ATMIS, we would recommend that they consider the guidance provided by Hopper [7]: "Look first at the information flow, and then select the right equipment to implement that flow. To use the best equipment to handle the most valuable information, you need to know which is the most valuable information." In the end, the objective is to produce an ATMIS design that will be responsive to the actual needs of the multiple users of information in the organization. This is more likely to result when the organizational elements, the required information, and the "equipment" (automated or manual) are integrated into a single system.

Development, Implementation, and Evaluation of the ATMIS

Once the final design is approved, a prototype of the new ATMIS can be developed and evaluated. In our view, the ATMIS should be developed on-site, if possible, in continuous consultation with the users of the new system and other technical experts, such as performance measurement experts and data analysts. The transition from design to development is then likely to occur with greater fidelity.

The main contribution that we offer regarding the test and evaluation plan for the ATMIS is that it address not only the traditional hardware and software performance criteria, but whether organizational performance is improved by the implementation of the new ATMIS. It is envisaged that precise organizational objectives be formulated during the ATMIS design phase and then incorporated into the evaluation planning process. These objectives, including performance criteria, would be derived by considering how the capabilities of the new ATMIS would improve organizational performance relative to that which was observed during the baseline condition. The extent to which the organizational objectives were actually accomplished, and how they were accomplished, would be assessed and documented during the evaluation period.

Aircrew training management information systems that are well-designed and implemented should be capable of improving organizational performance in numerous ways. We have attempted to make the point that the extent of the improvement will be a product of understanding the existing system as it functions within the organization, and coordinating new requirements with these kinds of analyses. Our goal is to improve organizational effectiveness and efficiency relative to the input of data into the system, its processing, and the output and use of information. Some examples of improved organizational performance include the following, which have been chosen to illustrate that improvements can occur at multiple levels throughout the training organization. 1) Aircrews no longer need to complete multiple forms which often contain redundant items of information. Instead, they can provide single, non-redundant items which are entered into the information system, and a computer program can automatically transcribe the data onto the required multiple forms. The administrative burden of the aircrews will, therefore, be made lighter. 2) Schedulers can now perform the job of scheduling more quickly or with fewer people, at the same level of effectiveness. 3) Needed analyses of training data can be completed, because the relevant data are collected

routinely as aircrews proceed through training programs. Automated collection and processing capabilities are also on-line to support these efforts. Previously, these analyses were rarely initiated, because they would have required small armies of people to tabulate and analyze the data contained on voluminous amounts of paper stored in file cabinets in academic and flying squadrons. 4) Reports are now used and in demand by the organization. This is because precise reporting requirements were ascertained, and the reports now contain relevant information in a useable format. 5) Key decision makers and their staffs can make important decisions in a reasonable amount of time, because they are now able to query an automated information system which contains valid training effectiveness data. They are able to justify expensive training system resources, such as flying hours and simulators, on the basis of this information.

CONCLUDING REMARKS

This final section is an attempt to summarize some of the major issues and factors which affect the design, development, and use of an ATMIS. Our primary emphasis has been on procedures for identifying the information required to support the training organization in accomplishing all of its major training and training management functions, throughout the projected life cycle of the ATS. This organizational perspective dictates that the ultimate users of the system must be key players and active participants in all phases of ATMIS design, development, and evaluation to ensure that their needs are met. The proposed approach is unique only in its organizational emphasis. The procedures discussed are consistent with the general systems engineering and Instructional Systems Development paradigms. The organizational orientation is crucial, however, to ensure that the ultimate value of the ATMIS is realized in terms of its payoff to the organizational users, and not in terms of its efficiency in processing enormous quantities of irrelevant data which can be distributed in large but otherwise useless reports. The remaining paragraphs identify some of the more important considerations and cautions which may contribute to successful ATMIS design and development.

<u>Recognition of value and organizational commitment</u>. The identification of the relevant information requirements for an optimal ATMIS is not a trivial task. Accordingly, all of the participants, particularly the using organizations, must be convinced that the value added is worth the considerable resource commitments that are required to ensure success. Unfortunately, we do not know of any clear examples of ATMISs which have been planned and developed using the information-oriented procedures recommended in this paper, and which can serve as exemplars. On the other hand, there are several examples of ATMISs which have been underscoped or have experienced other difficulties, because of the assumption that the ATMIS is primarily a data processing and hardware/software enterprise, which can be designed and developed in isolation from the broader organizational considerations and user needs.

<u>The risk of short cuts</u>. There are few, if any, short cuts in the proposed process that will not jeopardize the potential success of the ATMIS development. Especially crucial are: 1) the need for competent, representative user working groups to ensure a comprehensive, integrated approach to ATMIS design and development; and 2) the need to conduct a thorough baseline analysis to identify the formal and informal

information flow and processes at all levels of the using organization.

Acquisition approach. The traditional acquisition process requires that many things be set in concrete early in the development cycle. This is especially true in the case of fixed price contracts. As illustrated in the 93 BMW example, however, there is no reliable substitute for the labor-intensive analyses which are needed to identify the types of training information and the information flow of the organization. The resolution of this contractual issue requires a recognition of the potential value added by these analyses, as noted in the paragraphs above. Once this view is established, there are a number of ways to provide the contractual flexibility which is necessary to accomplish these kinds of analyses, including the use of a two-stage contractual approach similar to that employed in some of the current Air Force ATSs (e.g., SOF ATS).

Education. It should be recognized that as new, well-designed ATMISs come on-line, there is a need to educate users at all levels about the capabilities they can provide, and how they can best be used to meet their particular needs. This education process should be formalized initially to ensure that all relevant users are addressed. As experience is gained with the system's new capabilities, additional insights and side benefits will probably be identified. These should also be disseminated to users, in order to ensure maximum benefit to the organization.

REFERENCES

[1] Bruce, P.D., Killion, T.H., Rockway, M.R., and Povenmire, H.K. (June, 1991). B-52 and KC-135 Mission Qualification and Continuation Training: A Review and Analysis (AL-TR-1991-0010). Williams AFB, AZ: Human Resources Directorate, Aircrew Training Research Division.

[2] Fishburne, R.P., Spears, W.D., and Williams, K.R. (October, 1987). Design Specification Development for the C-130 Model Aircrew Training System: Final Report (AFHRL-TR-86-51). Williams AFB, AZ: Operations Training Division, Air Force Human Resources Laboratory.

[3] Rockway, M.R. and Nullmeyer, R.T. (in press). Aircrew Contracted Instruction: Major Issues and "Lessons Learned". Williams AFB, AZ: Human Resources Directorate, Aircrew Training Research Division.

[4] Dukes, R., Rockway, M.R., and Nullmeyer, R.T. (in press). Lessons Learned in the Development of the C-130 Aircrew Training System: A Summary of Air Force On-Site Experience. Williams AFB, AZ: Human Resources Directorate, Aircrew Training Research Division.

[5] Reakes, M. (1989). Requirements of a Training Management System for Aircrew Training. Proceeding of the 11th Interservice/Industry Training Systems Conference.

[6] Lucas, H.C. (1975). Why Information Systems Fail. New York, NY: Columbia University Press.

[7] Hopper, G. (May, 1991). "Prioritizing Information", Byte, Pp. 169-174.

[8] McGann, E.J.J., Rooney, J., and Rogers, M. (July, 1987). Reference Book for the C-130 Aircrew Training System. Little Rock AFB, AR: 34 Tactical Airlift Training Group.

[9] Bruce, P.D. (May, 1989). Aircrew Training Evaluation: B-52 and KC-135 Formal School Training (AFHRL-TR-88-49). Williams AFB, AZ: Operations

Training Division, Air Force Human Resources Laboratory.

ABOUT THE AUTHORS

Robert T. Nullmeyer is a research psychologist with the Aircrew Training Research Division, Armstrong Laboratory; Williams AFB, AZ. He is responsible for conducting aircrew training system research, which has focused primarily on training effectiveness assessments and training requirements analyses for Military Airlift Command and Strategic Air Command. Major efforts have included the C-130 Model Aircrew Training System study and several simulator training effectiveness studies with the C-130 weapon system trainer. Current efforts include a cost-effectiveness analysis of the C-130 ATS, developing training system evaluation concepts, and developing a methodology for assessing the effectiveness of mission rehearsal devices. He earned a B.A. in Psychology from Southern Illinois University-Edwardsville, and an M.A. and Ph.D. in Experimental Psychology from the University of New Mexico. He joined the University of Dayton Research Institute in 1978; and the Aircrew Training Research Division, Armstrong Laboratory in 1981.

Philip D. Bruce is a research psychologist with the Aircrew Training Research Division, Armstrong Laboratory; Williams AFB, AZ. He has conducted extensive field studies of Air Force and contractor-operated aircrew training systems in Strategic Air Command and Military Airlift Command. This work has included field studies of aircrew training management information systems, and a training effectiveness evaluation of the KC-135 weapon system trainer. His current research interests are developing methods for evaluation of aircrew training systems; designing management information systems; and the uses of aircrew training devices for combat mission training and rehearsal. He received a B.A. in Psychology from Indiana University, an M.A. in Experimental Psychology from Western Michigan University, and a Ph.D. in Experimental Psychology from the University of New Mexico. He joined the University of Dayton Research Institute in 1984; and the Aircrew Training Research Division, Armstrong Laboratory in 1989.

Marty R. Rockway is a Senior Research Psychologist with the University of Dayton Research Institute. After completion of undergraduate work in the physical and engineering sciences, he received a Ph.D. in Psychology and Mathematical Statistics from Northwestern University. He held a number of R&D posts within the Air Force Systems Command, including the position of Chief Engineer for Human Factors at the Aeronautical Systems Division, and he was Technical Director of the Technical Training and Operations Training Divisions of the Air Force Human Resources Laboratory. He has over 30 years experience in conducting research on training methods, media, and systems. He is currently involved in research to develop improved techniques for the analysis, design, evaluation, and management of aircrew training systems.

A Distributed Training System for Large Training Management Environments

Craig W. Shier
International Business Machines Corporation
Gulfport, Mississippi

ABSTRACT

Management of the training process from requirements definition through evaluation of graduates in the field is a complex process encompassing many specialists and organizations. A comprehensive training system is required which produces management information without increasing workload. The U. S. Air Force recognizes this requirement and is developing the Advanced Training System (ATS) to support every aspect of the Air Training Command's technical training mission. ATS will provide an integrated training environment across six Air Force bases with a potential for 35,000 user workstations and 175,000 students per year. ATS provides computer-assisted Instructional System Development and support for delivery of instruction for all media types. Computerized data collection supports timely access to student, resource, and evaluation information. New processor, software, and communications technologies provide a distributed training system with low administrative overhead. Configuration options allow equipment to be selected to meet the functional needs of individual users.

INTRODUCTION

The Advanced Training System (ATS) will support the technical training mission of the Air Training Command (ATC). ATS was motivated by a need to reduce the cost of developing, delivering, and maintaining training throughout the ATC. ATS is being developed in Ada under DoD-STD-2167A with significant Commercial-off-the-Shelf (COTS) software and exclusively on COTS hardware. With development beginning in May 1989, the project is currently in the Preliminary Design phase with Operational Test and Evaluation from January 1993 through May 1994. Life cycle considerations of using COTS (SW and HW) have been accounted for in the design process to ensure that ATS will be a state-of-the-art system when delivered. This paper summarizes the requirements and the preliminary design, and describes the lessons learned thus far in this system design and integration project.

FUNCTIONAL REQUIREMENTS

Two groups of requirements drive the design of ATS: (1) functional requirements which encompass cradle-to-grave support of training, and (2) System design constraints which derive from the need to support installation of up to 5,000 workstations at a single location without requiring mainframe computers.

The functional processes which ATS supports are Course Development, Course Delivery, Student Management, Resource Management, Training Evaluation, and System Management.

<u>Course Development</u> begins with the identification of a training requirement. Training managers and developers perform task analysis and follow the Instructional System Development (ISD) process as implemented by U.S. Air Force (USAF) and ATC regulations. A key design consideration for maintainability is that the ATS processes be designed according to the data requirements and not driven by the existing paper forms system. That is, the developer should focus on tasks, objectives, and other information products and not on filling out particular forms. A relational database will provide correlation of data elements within the system. Course Development supports ISD from the task analysis phase through the course validation phase.

<u>Course Delivery</u> manages the delivery of all courses whether they are lecture-based, computer-based, or a combination of delivery methods and media.

<u>Student Management</u> allows the registrar, student squadron, instructors, and other personnel to directly manage student information such as schedules, attendance, grades, and counseling data.

<u>Resource Management</u> tracks all resources needed for the training process from student quarters to classroom resources. Resource Management is integrated with the Course and Student processes to provide a comprehensive training management process which ensures that all planned courses can be successfully taught.

<u>Training Evaluation</u> involves collection of student and supervisor critiques of training, evaluation of graduates in the field, and verification that course data conforms to the standards imposed by regulations.

<u>System Management</u> provides on-line tools which minimize the skill level needed to operate and administer the system. It also requires a complete, 2167A-compliant software development facility (SWDF) in order to allow ATC to

maintain software after completion of the system development. Additionally, ATS will provide standard office automation tools such as publishing, word processing, spreadsheet and graphics support.

ATS will communicate to three external systems. The Occupational Measurement Squadron system will exchange task analysis data with ATS via the Defense Data Network (DDN). The Extension Course Institute at Gunter AFB will exchange Career Development Course materials with ATS using DDN. The Air Force Training Management System (AFTMS) will exchange class schedule and student accounting information with ATS via dial-up modem.

SYSTEM DESIGN CONSTRAINTS

Geography and life cycle considerations play important roles in a large system such as ATS. Together they drive the ATS system architecture. Geography affects the distribution of and access to information. It places constraints on the communications methods used to interconnect the computers which make up ATS. Life cycle considerations raise issues such as portability, maintainability, and training for ATS itself. To meet these constraints, it is important to consider what the marketplace offers in terms of products and standards.

Geographic Considerations

ATS must support installations of up to 5,000 workstations at each Technical Training Wing (TTW). Users must be able to log on and access data from anywhere in the system.. ATC HQ must be able to access information from the TTWs in such a way that the user does not need to know system information about the location of the information being requested.

TTWs are required to communicate via DDN. This network does not permit large amounts of interactive processing. Therefore, the system will have to route data and deal with integrity issues arising from transmission delays. Buildings within a TTW are required to communicate via Base Level Distribution System (BLDS) which is based on the Unified Local Area Network Architecture (ULANA). However, local conditions may dictate that some buildings run disconnected from the rest of the system. ATS must provide a mechanism for "connecting" these systems periodically through magnetic media. Intra-building communication between elements of the system is provided by ATS. To achieve compatibility with ULANA, Transmit Control Protocol/Internet Protocol (TCP/IP) and applications and communication components from the ULANA parts list will be used throughout ATS.

These geographic considerations motivate a distributed system with concomitant management, error handling, and data integrity considerations.

Life Cycle Considerations

Installation of ATS will take place over several years. The planned life cycle is 15 years. During this time period, the COTS equipment and software that will be available for new installations will change. Training requirements themselves will change as new training technology becomes available. These factors require that the ATS minimize the life cycle effects of portability, maintainability, and training to use ATS.

Life cycle considerations dictate that software be portable and as independent of specific hardware as possible. During the installation process, ATC plans to procure as much equipment as possible from USAF standard contracts. Product characteristics for future upgrades are difficult to predict. What is certain is that those products will increasingly adhere to standards that encourage portability and hardware independence and that these standards will evolve. Ada, POSIX, TCP/IP and eventually GOSIP, among others, will be at the core of new standards.

Maintainability considerations are a high priority because the installation of ATS will take place over several years, and the USAF desires to perform software maintenance functions rather than contract for support. ATS must not add a burdensome system administration organization to ATC. There will be only office type equipment and no raised floor environment. The USAF wants to maintain the ATS software through the 3302 System Support Activity at Keesler AFB. Use of standards increases maintainability and minimizes porting costs.

Since ATC's goal is to reduce training costs, ATS must not become a training burden in itself. Training for ATS will be self-teaching computer-aided learning. Task analysis and training development is being done in parallel with software development. When final testing is done, training will be validated in conjunction with evaluation of the entire system. Training requirements are being minimized through the use of a graphical user interface. Detailed operability standards are established to ensure maximum ease of use and commonality in Human-Machine Interface (HMI) design.

Marketplace Considerations

Today's marketplace is moving toward standards which will make it easy to meet the system and logistics requirements of ATS. Portability, heterogeneous distributed environments, and maintainability are all goals of these standards. In 1989, at the beginning of ATS, these standards were barely begun. Today they are maturing, but much remains to be done. The

successful system developer will anticipate the direction of the standards in his design. Definition and selection of COTS hardware and software requirements must be made with standards in mind. Flexibility must exist to upgrade products until late in the development process. Adaptation to the marketplace can be seen in the architecture development on ATS.

ARCHITECTURE

As mentioned previously, ATS is in the Preliminary Design Phase. The system architecture has been put in place and detailed requirements have been developed. Significant prototyping has taken place, especially in relation to Ada interfaces to COTS software.

The base contract does not assume any particular architecture or methodology beyond that implied by the system design requirements. The only initially planned COTS software packages were the operating system, database manager, and Ada Programming Support Environment (APSE). As the design progressed, COTS products were added for the office automation tools and courseware authoring.

Hardware Architecture

The hardware suite for ATS consists entirely of commercially available hardware. All equipment is compatible with a normal office environment. Compatibility with ULANA is achieved by using IEEE 802.3 connections between all processors, including workstations. To allow for flexibility and interchangeability in the marketplace, configuration items were defined in general terms; this also allows for hardware products to be selected and used in prototyping before committing to exact configurations or performance numbers. Installation specifications become the vehicle for describing a particular installation. The hardware was partitioned into five Configuration Items (CIs) as shown in Figure 1.

The Multiuser Processor is the host for developed Ada code and application data. The X-client/server model is used for workstation communication to the Multiuser Processors. This provides advantages in security, system administration, and software development.

The Workstation is an X-server. This can be either a specialized X-terminal or a DOS PC running an X-server emulator. In the case of the Computer-Based Instruction classroom, a DOS PC carries the additional ability to execute Authoring System development and delivery software.

The Communication suite contains requirements for LAN communications. ULANA components will be used to the extent possible to ensure design compatibility with the Base Level Distribution System.

The Printer suite and Scanner suite allow for several configurations: dot matrix to color laser printers, and graphics, text and optical mark scanners.

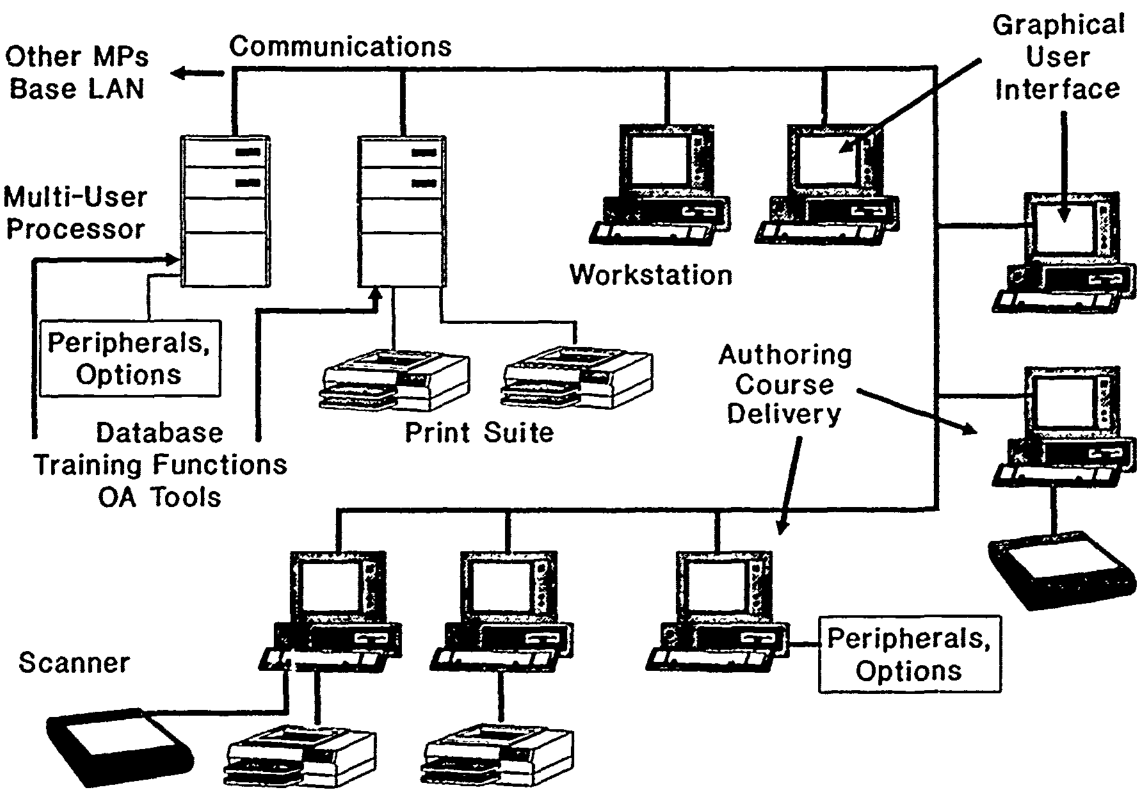


Figure 1. ATS Hardware

Software Architecture

Software was partitioned according to object-oriented lines. Each application Computer Software Configuration Item (CSCI) operates on a grouping of data. Applications are supported by two system-level CSCIs which isolate hardware dependent functions and distributed processing functions from the applications. Figure 2 shows the high level software architecture. At the lowest level, System Services provides the basic computer services, POSIX compliant operating system, relational database management, file management, and security on each processor. System Management provides application services which allow applications to be hardware independent and location independent. Applications use the System Management CSCI to exchange information through communications mechanisms provided by System Management. The System Management layer is critical to achieving portability. This is clear when the database architecture is considered.

ATS is a multiple database system rather than a distributed database system. Any given database resides completely at a node and is not distributed across several nodes. Databases are organized and distributed according to the needs of the organizational structure. System Management provides routing for application data requests. In order to assure portability, the application interface for such requests use only Ada constructs. System Management provides all calls to the database. System Management also provides interfaces to X-Windows to enforce human-machine interface standards. These interfaces are in the form of templates for predefined window types which applications access.

The Course, Resource, and Student application CSCIs map closely to the Course Development, Course Delivery, Resource, and Student processes identified above. They are user display functions which access the database through System Management CSCI. The Evaluation CSCI only needs to provide checklists and a reporting function. This is because the Evaluation process is facilitated by the structure imposed through the relational database by the Course CSCI. Much of the Evaluation process involves cross-checking various elements of the course control documentation to ensure its accuracy. This process is vastly simplified because the HMI checks user input to reduce errors at the outset. Additionally, the Course CSCI database ensures that traceability is maintained and eliminates the redundancy of forms that is inherent to a paper-based system.

Commercial-off-the-shelf Software and Prototyping

COTS software is being selected during the Requirements Definition and Preliminary Design phases. The selection process for each product involves a technical trade study and a cost-related tradeoff which involves licensing and life cycle requirements.

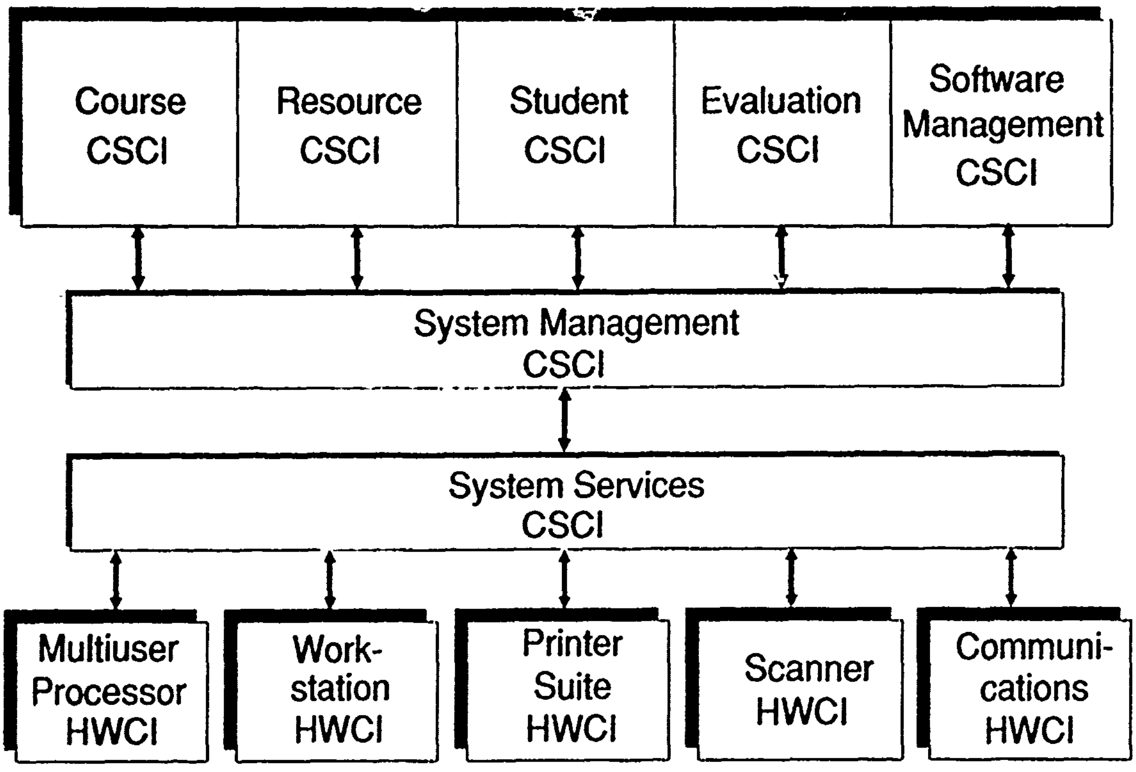


Figure 2. ATS Software

In using COTS for large functions it will be significantly more cost-effective to pay license fees than to maintain Ada code. However, given the installation and modification possibilities for ATS, the system design should have as much flexibility as possible in upgrades or modifications to COTS. Prototyping takes place during the trade study and during the early stages of design in order to ensure smooth integration while generating as few dependencies for the developed code as possible. Three areas of particular interest will be addressed here: Ada interfaces to X-Windows and Structured Query Language (SQL), Office Automation Tools, and Courseware Authoring.

Prototyping was critical to developing Ada interfaces to X-Windows and SQL. No standards exist for these interfaces. We contacted several organizations that had used Ada and X-Windows or SQL only to discover that each had to develop a unique methodology. The issues are basically that Ada and X-Windows contend over control of processing while Ada and SQL contend over data representations.

To ensure a useful design, two teams were tasked to develop prototypes. Working with X-Windows, the compiler, and database manager, interfaces were developed at two levels. (1). The Ada to X-windows and SQL interfaces were developed. (2). A higher level interface was developed for the application to access X-Windows and SQL. The two-layer approach is key to portability. In the first layer there are functions which are dependent on the database manager and X-Windows toolset. The second layer isolates the application from these dependencies. Therefore, if the X-Windows toolset changes or if the database manager is changed, no change to the applications will be necessary.

The Office Automation Tools will reside on the POSIX-based operating system on the Multiuser Processor. File access and distribution is controlled by the System Management CSCI through the RDBMS, but ATS treats output from Tools as BLOBs (Binary Large OBjects) and does not depend on the internal data format.

Authoring depends on a cooperative relationship between the Workstation and the Multiuser Processor. The Authoring systems that best fit the functional requirements of ATS execute on DOS rather than on UNIX systems which are POSIX complaint. The challenge was to use the power of the relational database to implement the ISD process on the POSIX MP and capitalize on the DOS environment for development and delivery of courseware without the security exposure that the DOS environment represents. The solution is shown in Figure 3. The Authoring system will execute under Windows 3.0. A An X-server will allow access to the Multiuser Processor for ATS application. The communication functions of TCP/IP and Network File System (NFS) will allow the Authoring system to create lessons and store them on the Multiuser Processor. These functions have been prototyped in such a way that the user need not be aware of the network functions which are taking place. The database manager will track files and correlate lesson files to objectives to preserve traceability in accordance with the ISD process. This also allows the Multiuser Processor to control file access and allows ATS to electronically distribute lessons to any classroom or delivery workstation. Additionally, student progress and scores can be stored directly on the Multiuser Processor, allowing easy access by the instructor supervisors, course evaluators, or administrative personnel.

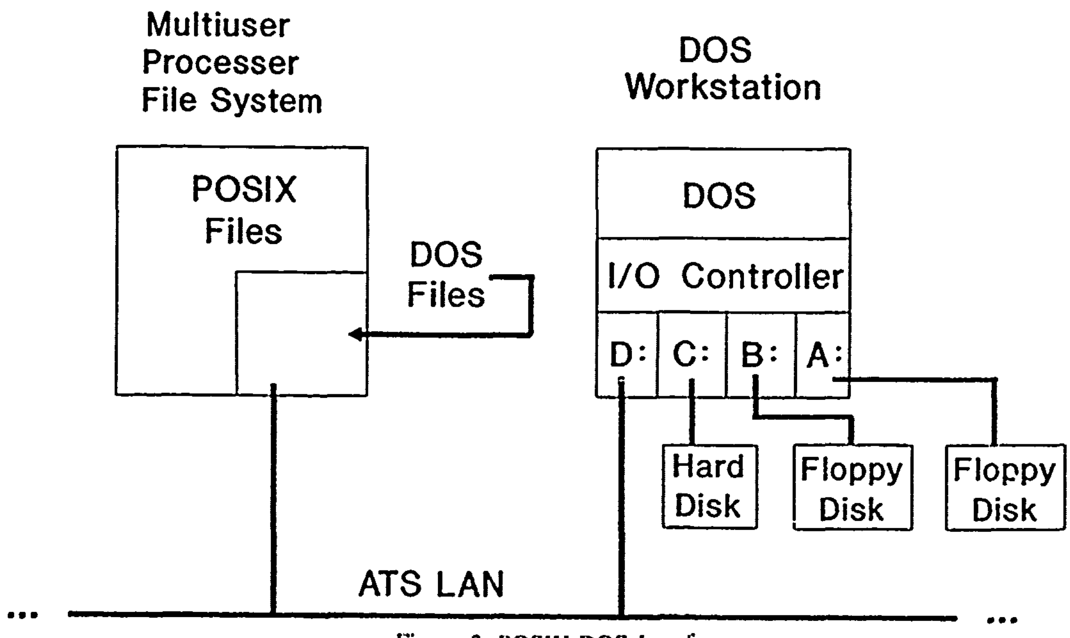


Figure 3. POSIX-DOS Interfaces

CONCLUSIONS

The ATS program has faced the challenge of developing a large distributed system in a time of rapidly evolving products and standards. The design process has capitalized on this change, turning it to an advantage by minimizing dependencies on changing elements and allowing later incorporation of COTS hardware and software products than is the normal practice. This advantage will continue throughout the life cycle of ATS by allowing the USAF to easily upgrade both hardware and software.

ABOUT THE AUTHOR

Craig W. Shier received a B.S.E.E. degree from Michigan State University in 1979. Since joining IBM in 1979 in Manassas, Virginia, he has worked in the areas of signal processing, sonar systems and combat systems development. His responsibilities have included technical positions as well as System Engineering Management. Since 1989, he has been the Chief Engineer for the Advanced Training System being developed in Gulfport, Mississippi.

THE MANAGEMENT IMPLICATIONS OF THE MODULAR SIMULATOR CONCEPT

by

James Brown
and
William Tucker
Boeing Aerospace and Electronics
Huntsville, Alabama

ABSTRACT

The Air Force has, with Tri-service support, contracted for research, development and demonstration of the modular simulator concept known as HAVE MODULE. Reactions to the concept, as developed by this program, have ranged from frank disapproval to open acceptance, but the most common is "What can HAVE MODULE do to help me with my problems?" In this paper,an attempt is made to answer this question. A dream of an ideal simulator development program is contrasted to often dismal realities. The contributions of the modular simulator concept that help achieve the ideal are discussed from a practical point of view, with emphasis on subcontracting. Some problems are described that the concept can help avoid and some that it will not. Lessons learned from the application of the concept to the demonstration, which was 75% subcontracted, and other projects in the specification stage are discussed. Recommendations are made for the future HAVE MODULE based programs.

INTRODUCTION

The HAVE MODULE architecture defines a logical grouping of generic simulator requirements into logical modules and defines interfaces between them. A standard hardware interface is provided if needed as in those cases where the logical modules might be assigned to physically separate computers, perhaps of different types. The requirements are grouped, as shown in Figure 1, to minimize coupling between modules, maximize cohesion within modules, and yet closely resemble classic simulator partitioning. The modules are defined in generic specifications and the data interfaces are defined in Ada language compilable constructs. Both are tailorable to any simulator program. This concept can strongly affect the way all phases of a simulator program are conducted. It is the opinion of the authors that careful application of the HAVE MODULE standards will ease each of these pha‘.s and reduce cost and schedule risk. I.. this paper we will present a dream of an ideal program and then describe how HAVE MODULE can help make this dream true.

THE DREAM PROGRAM

In the exciting time when a new contract is received, the front end analysis needed to prepare a strong technical base is often bypassed in the haste to "get on with the program". Long lead parts need to be procured, subcontracts let, drawings released, and software design started. There is no time to do esoteric functional analysis or preparation of detailed interface specifications. We start this job like we have all the others, knowing that there will be integration problems when the suppliers deliver, but rightfully confident that our good engineers will solve them. After all, they did last time.

But this time, we tailored the generic HAVE MODULE specifications to our application during the proposal phase and used them in our supplier negotiations. We updated the interface definitions during technical negotiations and included them in the agreement. Strangely, after contract award we're still busy, but the procurement people complain less and the software manager doesn't show up to complain about systems engineering quite so soon. Soon we finish releasing our requirements and go into the next phase.

Development

As the software designers attempt to implement systems engineering's usually cryptic requirements, they are tempted to once again allow their imagination free rein to design the system they like, but find themselves bound by the Ada language interface definition and the detailed functional specifications. From time to time they gleefully report an interface exception to the program office, but are quickly provided a repaired coordinated interface. When they exercise the static and dynamic test cases provided with the specification, they realize that they have no choice; they have to meet the requirements. They are comforted by the realization that the models they have now turned over for integration are similar to what they have done before and will be a good contribution to their store of off the shelf routines.

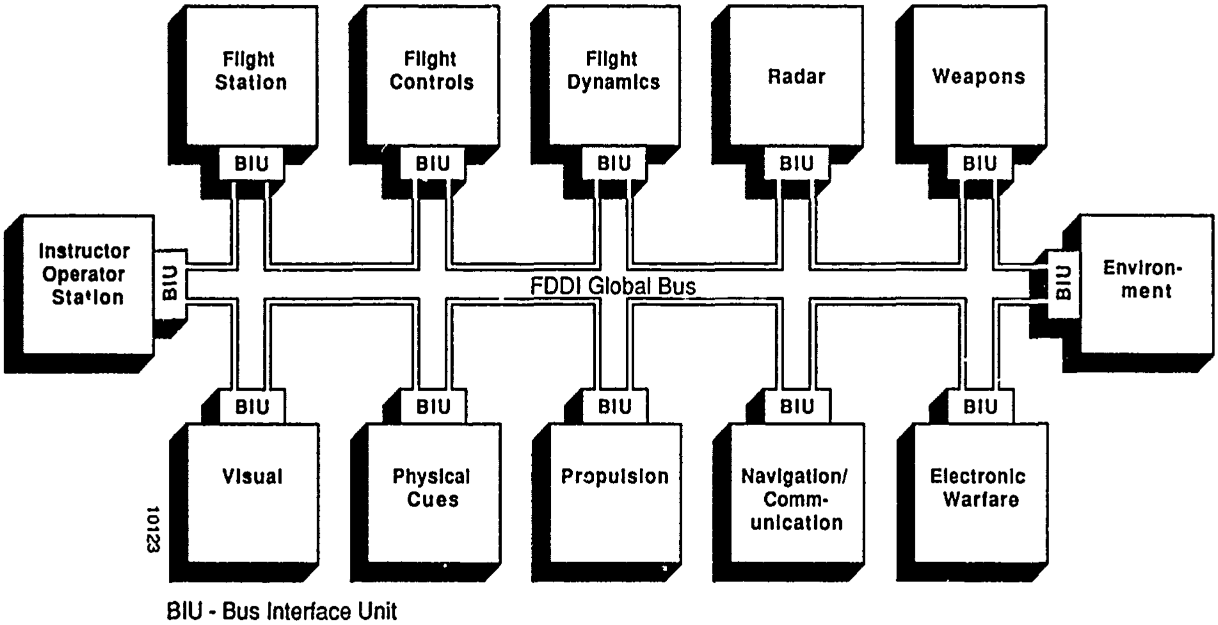


FIGURE 1
Modular Simulator Architecture

<u>Integration</u>

Now is the time for the integration engineers to shine. All of the suppliers and in house builders have passed their ATPs and all of their subsystems are ready to play together. In the bad old days, now we would start wishing we had done better systems engineering. But, since we published our functional and interface specifications at contract award, maintained them through development and enforced them at supplier acceptance, we suddenly experience instant integration. Shocked, we begin test.

<u>Test</u>

After all the successes we have had in our ideal program, we enter into test confident that we will again have a record setting success. However, success at this point is more a function of how well we allocated the top specification requirements, how well we met them, and how well we designed the test procedures to prove them. However, we will assume that the systems and test engineers used some of the time and effort they saved at proposal time and integration time to meet these goals. Therefore our testing proceeded at the same startling pace as our previous efforts.

<u>Conclusion</u>

Now, congratulated by our peers, rewarded by our superiors, and appreciated by our customer, we look back at our ideal program. We realize that we've been dreaming after falling asleep reading a boring RFP. None the less, we plan to closely examine the concept to gain some of these promised benefits. At the office the next morning, resolved to learn more about this approach, we drop by our friendly local HAVE MODULE ISWG representative whose presentation precipitated our dream. When the representative ran down after talking about loosely coupled standard modules and global data busses, we asked "What does that mean to me ?"

"Basically," said our loyal representative, " HAVE MODULE provides you with a standard simulator architecture, comprising twelve separate logical groups, documented in a set of specifications. Included in the specification is a generic interface description, written in Ada, and tailoring instructions which describe how to adapt the provided specification and interfaces to your simulator. Test software is available to test the completed modules along with software tools to manage the Ada language interface and keep the Bus Interface Unit software and test software current with the released interface definition."

"The specification has been successfully adapted to two programs separate from the demonstration project. The lessons learned from those adaptations along with experience gained in the demonstration project is implemented in the tailoring guide as down to earth practical instructions on how to modify the requirements to meet a particular need. The support software is available in an "as is" condition from the Air Force. Use of the software system with a limited set of modules (multiple segments in a module) and limited module sets (like systems without weapons or radar simulations) have

been proposed. Further study is required to validate the use of the concept with multiple crew station simulators. A concept has been developed and documented to interconnect devices using the Distributed Interoperability Standard (DIS) Protocols. Figure 2 shows the research status in these areas."

Technology Goal \ Program Component	Conceptual Design	Generic Specification	Tailoring Guide	Demonstration	Independent Verification	
Baseline Partitioning, Interfaces, and FDDI Bus	√	√	√	√	√	
XTP Protocol	√	√	√			
Limited Module Set	√	√	√	√		
DIS Interface Module	√	√				
Multiple Logical Segments in a Physical Module	√					
More Fully Functional F16C	√	Δ				
Multiple Crew Stations						

Δ Application specific specification

Figure 2
Have Module Research Status

THE REAL PROGRAM

The research that has been done has demonstrated that benefits are available from the HAVE MODULE concept as shown in Figure 3. These benefits were demonstrated not only by the HAVE MODULE program, but by two independent agencies who tailored the generic system/segment specifications for their particular applications. Phil Peppler [1] of Williams Air Force Base Armstrong Laboratory, and Terry Snyder [2] of Grumman's Simulation/Training Programs provided their experience with tailoring the specifications during the sixth HAVE MODULE Interface Standards Working Group. The results were: The HAVE MODULE architecture allows for a straightforward design and development and is complementary with Ada; It allows engineers to focus on the simulation requirements rather than "specmanship"; It encourages reusable modular designs; It does not force a particular design approach ,but allows design to be determined by analysis, judgement,and resources and; It was a good vehicle for flowing down requirements. The HAVE MODULE concept has also been selected for use on the Army's Advanced Distributed Simulator Technology program.

Tailoring these specifications to provide module level requirements and interfaces at proposal time will greatly ease the pain of program startup by providing a stable engineering baseline. This has always been a goal but the relative ease to tailor generic specifications makes it an achievable goal.

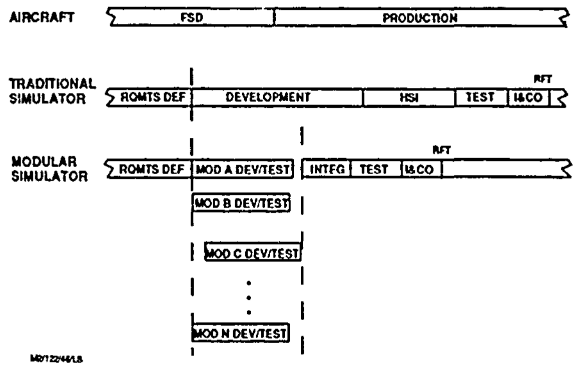


Figure 3
Representative Have Module Schedule

Development

When the development phase starts with well defined requirements and interfaces, the application engineers can focus on the application software design. We found this phase of the program to contain the most critical need for interface meetings/telecons between the module designers. During the HAVE MODULE demonstrator development, the module designers identified a very small number of interface changes. Because the HAVE MODULE concept extends tight interface control deeply into the program from the very beginning, a required revision to the interface must be developed and coordinated very quickly. The BIU and the Module Tester software must be revised and recompiled and distributed to the module designers in a very short time. Therefore, special software tools were developed to allow very rapid coordinated changes to be made to the interface. Because each change required each module builder to recompile his load, changes were scheduled for block updates. Other than the need to continuously coordinate the interface, the HAVE MODULE concept allows for highly parallel and independent individual development and test of individual modules.

Module Test

The module test phase is the most critical phase of this type of program. A principal advantage of the HAVE MODULE concept is that once a module has passed module test it will integrate smoothly into the system with few problems. The thoroughness of module testing is directly proportional to the ease of integration. This fact became painfully obvious during the integration of the demonstration device. Some modules required tuning (flight dynamics) and rework (IOS and weapons) during integration. In retrospect, the test cases for those modules were not thorough enough to adequately test the modules. Test data may not be available at the proper level to qualify a given module, but if the mode and state transitions are tested along with logical extremes and representative worst cases, integration will occur smoothly.

Integration

Integration of successfully tested modules can be a rewarding experience. By properly scheduling the modules, you can enter integration with confidence that the modules will communicate without problems. This allows you to focus on typical simulator tuning problems sooner. Even though the HAVE MODULE demonstration program experienced more difficulties than anticipated, integration was still successfully performed in much less time than a typical simulator program (18 days).

Test

At the system level, test of a modular simulator is no different than any other simulator test. However if system level requirements were adequately assigned to the module level and properly tested, the system testing will be painless. One key advantage of the modular architecture is that, once a problem is isolated to a module, that module can be pulled off line from the simulator and placed in a module test mode for troubleshooting while the rest of the device continues integrated testing of unaffected systems.

To assist this isolation process, the HAVE MODULE program created software integration tools that capture messages and data on the FDDI bus. Additional tools such as a FDDI Bus Analyzer and an enhanced IOS data display pages would be very beneficial. A real time debugger would be essential to quickly troubleshoot problems in a deliverable trainer environment.

Using the existing software test tools and the additional tools mentioned above, testing will require less time than traditional simulator programs.

CONCLUSION

The HAVE MODULE architecture can help a simulator development program to reduce cost and schedule and improve supportability by lowering technical risk. It does this by providing an early firm specification and interface baseline readily derived from an easy to tailor generic specification.

REFERENCES

[1] Peppler, Phil Presentation at The HAVE MODULE ISWG #6 July 11-12, 1990.

[2] Snyder, Terry Presentation at The HAVE MODULE ISWG #6 July 11-12, 1990.

ABOUT THE AUTHORS

Mr. James Brown served as the engineering subcontracts manager for the modular simulator demonstration program. He has also performed as engineering subcontracts manager for the United Kingdom E-3 simulator program and on various avionics and weapon systems programs. He received his Bachelors degree in electrical engineering from the University of Arkansas in 1986.

Mr. William Tucker was the project manager on the demonstration project. He has also been project manager for the KC-135 production contract, the UK E-3 project and other programs. He has worked in simulator software and systems design since 1977 after serving two years as a field artillery officer in the US Army. He received a Bachelors degree in electrical engineering from Wichita State University in 1975.

EMPOWERMENT: A MODEL FOR MANAGEMENT ACCOUNTABILITY

Dr. Graydon Dawson,
Courseware Product Assurance
Simms Industries, Inc.
Subcontractor to
McDonnell Douglas Training Systems'
C-17 Aircrew Training System
Norman, Oklahoma
[405] 366-5526

ABSTRACT

Empowerment is a critical component of a Total Quality Management (TQM) system. Total Quality Management training that has been the most successful include a paradigm-shifting set of experiences for the managers in training which are, in turn, transferred to the job resulting in a highly effective and empowered work force. How many managers in your organization have a working paradigm that is consistent with the principles of TQM? What is your organization doing with and for the other managers who's paradigms are not working? Effective TQM training addresses, head-on, the managerial habits (paradigms) that are counter-productive to effective TQM. An effective model of management accountability will include performance standards - the characteristics of a paradigm in harmony with the principles of TQM, and a measurement tool for measuring whether a manager's paradigm is moving (shifting) towards empowering their work force. Conclusions from one year of tracking and reporting manager's empowerment behaviors, at McDonnell Douglas' C-17 Aircrew Training System Courseware Development site in Norman, Oklahoma, will be drawn. Successful and unsuccessful empowerment strategies used by **Malcolm Baldrige National Quality Award** winners and non-winners will also be reviewed.

INTRODUCTION

More and more organizations are allocating resources for the development and implementation of Total Quality Management (TQM) training. The reason is that organizations which have been highly successful in the market place (i.e., Malcolm Baldrige National Quality Award winners) have exhibited a commitment and practice of empowering their work force. Arguably, trying to copy organizations that have won the Baldrige award may be the wrong motivation for implementing TQM. More and more organizations are getting more press/attention because of their emphasis on TQM training. Unfortunately, far too much of TQM training is nothing more than a quick fix and the outcomes are short-lived, short-term and cosmetic.

Empowerment is a critical component of effective TQM implementation. Management's role in empowering the work force is to provide leadership in, and the necessary resources for, establishing organizational structures (models) of responsibility, accountability, and authority. Unless management has established organizational structures that include: 1) performance standards for empowerment, 2) training, 3) how to measure performance against standards, 4) how to use diagnostic/prescriptive feedback, and 5) the rewards for attaining performance standards and the consequences of not, organizations will never realizes the fruits that successful organizations are realizing through their TQM implementation.

The purpose of this presentation is to demonstrate that TQM training programs that have been the most successful establish management structures of accountability and include a paradigm-shifting (*) set of experiences for the managers in training which are, in turn, transferred to the job resulting in a highly effective and empowered work force. The characteristics of a paradigm in harmony with the principles of TQM will be defined and a measurement tool will be reviewed for measuring whether a manager's paradigm is moving (shifting) towards empowering their work force. Conclusions from one year of tracking and reporting manager's empowerment behaviors will be drawn and we will also examine both successful and unsuccessful empowerment strategies used by Malcolm Baldrige National Quality Award winners and non-winners.

THE COMPONENTS OF A PARADIGM IN HARMONY WITH TQM

Adams and Kinchen (1990) define Total Quality Management as "a customer-driven operating philosophy committed to excellence in products, services and

relationships through total participation in the constant improvement of all processes." [1] Adams and Kinchen, have spent the last ten years implementing TQM within both large and small organizations and have drawn some significant conclusions (lessons learned).

Although successful TQM implementation is as unique to each organization and management system as personal development is to us as individuals, one common feature that Adams and Kinchen's research has born out is that "everything happens, or doesn't happen, on the basis of relationship." [1] TQM training that focuses on how to cultivate an effective relationship between a manager/supervisor and his/her people provides the glue which is critical to holding the various components of TQM together. Quality Circles, Participative Management, Pep Talks, Statistical Process Control, Suggestion Systems, Flattened Organization, or Team Building activities have all proven to be quick fixes unless they are built on a foundation of a strong and effective relationship between manager and employee.

A study conducted by Harbridge House (1984), a Boston consulting firm, identified ten managerial habits which profile this foundational relationship between manager and employee:

1) Provides clear direction, 2) Encourages open communication, 3) Coaches and supports people, 4) Provides objective recognition, 5) Establishes ongoing controls, 6) Selects the right people to staff the organization, 7) Understands the financial implications of decisions, 8) Encourages innovation and new ideas, 9) Gives subordinates clear-cut decisions when they are needed, and 10) Consistently demonstrates a high level of integrity. [2]

As Adams and Kinchen (1990) point out, "successful TQM implementation involves changing very long-standing and deeply entrenched organizational and managerial habits. The implications of these changes are shifts in power, authority, communication patterns, performance evaluations and the basis for promotions, just to name a few." [1] All the grassroots enthusiasm in the world is not enough to effect lasting change without a fundamental change in people who hold positions of status and power (managers and supervisors). And let's face it, most of what we experience in the form of management training today simply does not take on these difficult areas of training because it shakes the very foundations and assumptions about human motivation in the work place upon which most corporate cultures and structures are built.

Stephen R. Covey (1989), author of the inspiring national best seller <u>The 7 Habits of Highly Effective People</u>, discusses the power and importance of paradigm shifts to effective interpersonal relations. Each of us have an operating paradigm - a psychological map, a personal frame of reference, the way we see the world - not in terms of our visual sense of sight, but in terms of our perceiving, understanding and interpreting relationships. All the influences in our lives all have made their silent unconscious impact on us and have helped shape our frame of reference, our paradigms, our maps. Furthermore, our paradigms, correct or incorrect, are the source of our attitudes and behaviors and, ultimately, our relationships with others. And frankly, as Covey purports, most of us who are managers need a paradigm-shifting experience in order to be more effective in relationships. [3]

To help us see more clearly what Covey means by a paradigm-shifting experience, he cites a number of examples. The first is the story about a man who was reading a book on a New York subway train. Several other passengers were passing the time reading their daily newspapers when a young father with three children boarded the subway car. While the young father sat staring at the floor, his three children wreaked havoc with the passengers, chasing each other back and forth, wrestling each other, knocking the newspapers out of the hands of the other passengers and, in general, upsetting everyone except the young father. Covey, who was himself observing this situation, couldn't help wondering how this young father could be so oblivious and insensitive to the chaos his children were creating. Covey thought surely this young father will notice what's happening and discipline his kids. But that never happened and most of the passengers' non-verbal behavior suggested that it was the father who needed to be disciplined. His patience wearing thin, Covey went over and sat next to the young father and pointed out that his children were out of control and asked if he could not see that? The young father looked up from the floor to see the faces of the passengers frowning and responded, "Oh, you're right. I guess I should do something about it. We just came from the hospital where their mother died about an hour ago. [illegible] don't know what to think, and I guess they [illegible]n't know how to handle it either." Covey says, can you imagine what he felt at that moment? His paradigm shifted. Suddenly, he saw things differently, and because he saw differently, he thought differently, he felt differently, and he <u>behaved</u> differently. His irritation vanished. Everything changed in an instant.

A more glaring example of how one's paradigm determines how they see, and subsequently interact, with the world (so-to-speak) was the paradigm shift Ptolemy must have experienced. "For Ptolemy, the great Egyptian astronomer, the earth was the center of the universe. But Copernicus created a paradigm shift for the followers of Ptolemy, and a great deal of resistance and persecution as well, by placing the sun at the center. Suddenly, everything took on a different interpretation."

What is the substance of a paradigm shift that must take place in the hearts and minds of managers?

William C. Byham (1989) authored a simple, yet powerful book entitled Zapp! The Lightning of Empowerment. [4] In Zapp! we see the daily transformation (paradigm shift) of a supervisor (Joe) as he learns that continuous improvement for the individual and the company is based on the relationship between himself and the employees who report to him. Joe learns that his basic assumptions about how to motivate his people (e.g., managerial habits) have been acquired by watching other managers. These habits, Joe learns, have created a working environment that seemingly builds mistrust and apathy among his work force.

Joe's initial paradigm manifests itself in taking responsibility away from his employees, taking away employee authority to make any decisions that affect the employees performance, and taking away the employee's identity, energy and power. However, Joe learns the value of having an effective role model when he observes another manager giving her employees responsibility, authority, identity, energy and power. Subsequently, his paradigm begins to shift. Joe learns that sharing these critical elements of human motivation with employees does not mean that he is abandoning any responsibility. Joe begins to question the role model he is providing for his employees and learns that the model his employees observe is largely responsible for the quality of the working climate.

The Harvard Business Review (1987) reports that 60 to 70% of the climate in which we work is credited to our manager. [5] The real tragedy acted out in most work places is that millions of people are allowing themselves to be treated with something less than human respect because they are afraid to risk objecting to it. Then, when they themselves become supervisors or managers of other people, they are often just as insensitive as their bosses. After all, they see that kind of behavior being rewarded all around them.
Joe, our supervisor in Zapp!, learns about a force which energizes his people, helping employees take ownership of their jobs so that they take personal interest in improving the performance of the organization. Joe learns about four categories of management behaviors (habits) which Zapp! his people: 1) Maintaining the self-esteem of his people, 2) Listening and Responding with empathy, 3) Asking for help in solving problems, and 4) Offering help without taking responsibility. Joe learns that these behaviors are his responsibility for initiating and maintaining. Furthermore, he learns that he must first empower individuals before he attempts to create empowered teams. That is, the individual employees' personal experience with empowerment must be cultivated and understood before a team can be effectively formed and transformed. Zapp! builds a powerful case against most TQM training programs which fail to include a personal transformation - changing very long-standing and deeply entrenched organizational and managerial habits. How refreshing and uplifting it is to find a manager who, not only gives verbal ascent to TQM, but can back it up with very effective relationships with his/her people in establishing a working environment where anyone can question any system or process. TQM training which seeks to set up empowered teams within work groups without first getting the manager/supervisor to analyze their own interpersonal relationship (managerial habits) with their people offer little more than quick fixes.

Our character, basically, is a composite of our habits. Habits are behavior patterns and thought patterns which get repeated so frequently they become automatic, conditioned responses to behavioral triggers or situations in which we find ourselves. The conditioning of a lifetime affects every manager's perceptions, how they see things, their attitudes and the way they interact with other people. Is it possible that some managers are operating with an ineffective paradigm (not conducive to TQM precepts)?

Covey compares our paradigms to a road map and raises the question "how useful would a road map of Detroit be if we wanted to get to a specific location in central Chicago?" Can you relate to the frustration, the ineffectiveness of trying to reach our destination? We might work on our behavior by trying harder, being more diligent, double our speed, but our efforts, Covey says, would only succeed in getting us to the wrong place faster. Or, we might work on our attitude by thinking more positively. The point is, we would still be lost because the fundamental problem has nothing to do with our behavior or our attitude. It has everything to do with having the wrong map. Covey says that the power of a paradigm shift is the essential power of quantum change, whether that shift is an instantaneous or a slow and deliberate process. [3]

How many managers in your organization have a working paradigm that is consistent with TQM precepts? What is your organization doing with/for the manager whose paradigm is not working? Effective TQM training addresses, head-on, the managerial habits (paradigms) that are counter-productive to effective TQM with training, self-assessment and retraining. Ineffective TQM training avoids these issues.

Empowerment (training and practice) is not a quick fix or personality technique we put on like a coat in order to be more effective with people or be more liked by people. Covey goes on to say that the

requisite paradigm-shifting experience must first be a private victory, based on thorough self-analysis, before it can be a public victory. It is futile to try to improve relationships with others (quick fix techniques) before we improve ourselves. Doing what we have been doing over and over and over again but expecting different results from our work force is a definition of insanity. Most management training, however, does not address the private victory in the TQM equation.

Ralph Kilmann (1987), author of Beyond the Quick Fix, states that each organization has five leverage points (tracks) that can affect morale and performance: 1) the organization's culture, 2) the manager's skills for solving complex problems, 3) the group's approaches to decision making and action (team-building), 4) strategic choices and structural arrangements, and 5) the purpose and design of the reward system. More importantly, Kilmann states that these five tracks require months for planning and implementation and are sequential in their implementation, each building upon the preceding track. [6] For an organization to channel its resources/programs in any of these tracks without first having built upon the preceding track is to allocate/channel resources for quick fixes, according to Kilmann. Critical to building a strong TQM foundation is a thorough review and analysis of the organization's culture (guiding and operating values) with management trainees followed by the requisite management skills training to facilitate the new corporate culture.

One of the major areas in which a paradigm-shifting experience must take place is in a self-assessment of our assumptions concerning human motivation in the work place. Research by Herzberg (1987) has been replicated by numerous other studies with the same conclusions. [5] There seem to be two categories of human motivators. Furthermore, effective organizations/managers apply them in working with their people.

Herzberg calls the lower level motivators "maintenance factors" or those that must be present to maintain a minimal level of satisfaction (e.g., job security, salary, work conditions, company benefits and policies). However, higher levels of satisfaction can only be realized when the employees are provided opportunities (empowered) to use their intellect to improve the process. Without involvement, there is no commitment. Effective organizations and their managers are paying more attention to these higher level motivators by developing systematic programs or personal habits so that employees can experience achievement, recognition, advancement, be turned on by the work itself and experience personal growth that comes with increasing levels of responsibility (i.e., Employee Motivation + Empowered Opportunity + Achievement + Rewards = Employee Commitment). The winners and non-winners of the **Malcolm Baldrige National Quality Award** have realized these important keys to success.

The Juran Institute, Inc. (1991) has been close observers of the Baldrige Award winners and non-winners in terms of the things they did to achieve stunning quality results. For the winners and non-winners, quality results were a result of a combination of strategies employed and not one here and there. According to the Juran Institute, what did work was: 1) processes at the worker level were revised by the workers so as to put workers in a state of self-control, 2) the work force were provided opportunities to participate actively on quality improvement teams, and 3) test sites were established at which teams of workers were trained and empowered to become self-supervising. [7] In general, these organizations put into practice the concept that planning for quality should involve participation by those who will be impacted by the plan.

What did not work for the Baldrige winners and non-winners, according to the Juran Institute, was: 1) Massive meetings of employees, speeches, wall posters, pledge cards, slogans, and the colorful rest. Such spectacles lacked substance, and were commonly views as "here comes another one." Subsequently, the credibility of the sponsoring managers was reduced, 2) programs whose sole emphasis was on Statistical Process Control. Organizations which focused on training in tools, alone, generally were focusing on the useful many improvements while neglecting the vital few, 3) Quality Control Circles, 4) the Project-by-Project approach to improvements neglected the vital establishment of a corporate quality infrastructure which harness and focus all quality initiatives and resources, and 5) Increased inspection and testing. [7]

Many valuable lessons can be learned, and mistakes not duplicated, if managers would take the time to seek out research findings such as these. Unfortunately, according to McGregor (1960), most managers reject the findings of social science research. Instead they believe that their own experience is an adequate database on which to make decisions. [8]

Covey (1989) reminds us of Aesop's fable of the goose and the golden eggs - a story about a poor farmer who becomes fabulously wealthy when he discovers his pet goose (the production asset/capability) lays glittering golden eggs (production). However, with his increasing wealth, comes greed and impatience for more and he kills the goose to get all the eggs at once. He not only discovers no eggs, but the producing asset (the goose) is no longer capable of producing any more of the

prized golden eggs. Covey suggests that there are three kinds of assets in every enterprise - Physical assets, Financial assets and Human assets. Our poor farmer, and far too many managers, place all their eggs in one basket - Financial Assets (e.g., Quarterly Reports, decisions driven by production quotas). However, it is the human assets that have control over both the physical and financial assets. If managers operate from a paradigm that focuses on golden eggs and neglects the care and feeding of the goose, we will soon be without the asset that produces golden eggs. When managers fail to respect the Production/Production Capability (P/PC) balance in their use of physical and financial assets in organizations, they decrease organizational effectiveness. [3]

One of Edwards W. Demings' 14 points is to "give people an opportunity to take pride in their work." Deming says that managers in the U.S. do not utilize what they have available to them - the creative minds of their people. In fact, Deming says, when it comes to utilizing the intellect of its own work force, the United States is a third-world country. If management did a better job of utilizing their people, says Deming, they would get a higher level of employee commitment.

McGregor says, that "commitment is a function of the rewards associated with achievement." [8] No doubt you've heard it said that management gets the behaviors it rewards and the behaviors that are not rewarded, go away. Employee commitment is a function/outcome of: 1) providing employees opportunities which facilitate the higher level motivators and 2) providing a reward system that is tied to the higher level motivators. And management controls both of these factors in our equation - the giving of opportunities (empowering) and the rewards based on achievement. And the personal paradigms that far too many managers are operating with are bent towards controlling, repressing and intimidating their people.

A DEFINITION OF EMPOWERMENT

A foundation has now been laid on which a definition of empowerment can be built. Empowerment is the effective application of understanding, enabling, and encouraging our people for the constant improvement of all processes.

Understanding our people means, as manager, we must possess an operational knowledge of the research on human motivation in the work place. We build a much more solid foundation on which empowerment is defined, and measured, when we managers understand and apply what research tells us about the factors that enhance employee commitment, loyalty and interest in improving the job processes, products and services.

Enabling our people means we give our people opportunities to realize the higher-level motivators, take ownership of their jobs, operate within a team structure for the purpose of continuous innovation and improvement of all processes. Measurement systems (quality tools/techniques) provide a feedback system with indicators of how well our processes are performing to meet our internal and external customer requirements (satisfaction).

Encouraging our people means we maintain their self-esteem (e.g., When we create value for other people, they soon create value for us), we listen and respond with empathy, ask for help in solving problems, and offer help without taking back the responsibility and authority. Understanding, enabling and encouraging our people represents the heart of a customer-driven operating philosophy committed to excellence in products, services, and relationships through the total participation in the constant improvement of all processes.

WHY AREN'T LEADERS LEADING?

A Gallup Poll surveyed 401 CEOs of America's largest corporations. The results indicated the following: 1) Most CEOs know that American firms have a problem with quality, 2) Over 50% of the CEOs surveyed said that they did not accept responsibility for problems associated with quality, 3) Over 50% of the CEOs said that it is the employees' lack of skills, commitment and understanding of their work that makes it difficult to deliver a quality product or service, 4) 61.7% of the CEOs said that the lack of management attention to quality does not affect quality, and 5) 70% of the CEOs said the pressure for short-term profits did not have an impact on quality. [9]

William Roth reports that the long-term objective in investing in quality improvement is to steadily improve the corporation's bottom line through better planning, relevant training, the introduction of appropriate statistical measurement tools and better use of employee expertise (empowerment). Short-term objectives have been to enhance the company's image by publicizing its' new dedication to quality improvement. This is evident by senior managers who appear periodically to make well crafted speeches which often note that improved quality requires "cultural change and must become a way of life." The trick, as Roth reports, is to watch their feet as well as their mouths. Imaging is easier than doing. And because it is easier, senior management becomes more interested in creating the image of improved quality than in actually improving it. According to Roth, "Employees learn all too frequently, that upper level managers are indeed for improved quality and the necessary changes, but only so long as they themselves are not affected and only so

long as alterations in their own style of management (paradigm) are not necessary. They are currently involved in too many crises upon which the fortunes of the company depend to worry about changing the way they do things. If the top people don't set the example and play by the rules, no one else will. If the top people decide they are allowed to modify the rules to deal with the pressures of leadership others will quickly follow suit." [10]

Deming says that 94% of the quality problems in most organizations can be traced to problems in the organization's own systems and process, whereas, only 6% of the quality problems can be traced to a particular employee. Who owns the organization's systems and processes? Managers are the only one's given the authority to allocate/approve resources to make improvements/changes to the organizations processes/systems. How can the system/processes in an organization be improved/changed? Only through the intellect of the work force. And yet, how do managers in most organizations get ahead or climb the corporate ladder? By conforming to the system rather than by changing/improving the system. Do you **get a sense of the dilemma most organizations face when trying to move past making verbal commitments to TQM to a legitimate operating philosophy that effectively empowers its work force?**

The case presented thus far begs two questions, 1) What is an effective paradigm - one in harmony with the precepts of TQM? and 2) How can a manager determine (measure) whether his/her working paradigm is moving (shifting) towards empowering their work force?"

An effective paradigm reflects a balance between production (performance) and the production asset/capability (the care and feeding of our human assets). The guidelines for human conduct that are proven to have enduring, permanent value are fairness, integrity, honesty, trust, human dignity, a servant's attitude/practice, an urgency for quality, being personally responsible for cultivating the potential of our work force through training, retraining and nurturing opportunities to be empowered, patience with people and a tolerance for mistakes, and daily encouragement. These guidelines for human conduct (leadership behaviors) form the basis for a paradigm consistent with the precepts of TQM. How can we establish a structure for management accountability where these Empowerment standards of behavior can be measured (baselined), reported, and the results used for constructive purposes?

MEASURING AND REPORTING EMPOWERMENT BEHAVIORS

A familiar continuous improvement axiom says, "To improve anything, you have to have a baseline." That is, we cannot know whether or not we have improved until we can compare where we are to where we were (baseline) or where we should/would like to be (some standard of performance). The Empowerment Indicator Survey which follows measures four categories (subscales) of management-employee relations which management is responsible for initiating and maintaining. This survey was developed by this author (after reading William C. Byham's <u>Zapp! The Lightning of Empowerment</u>) and is currently being used with McDonnell Douglas Training Systems courseware development managers for the C-17 Aircrew Training System in Norman, Oklahoma.

After reading <u>Zapp! The Lightning of Empowerment</u>, all employees complete the Empowerment Indicator Survey rating the supervisor or manager to whom they report. The Empowerment Indicator Survey is an indicator of: 1) The degree to which the delegated responsibility, accountability, and authority of empowerment is being realized by the people who report to the manager or supervisor being rated, and 2) The degree to which the responsibility, accountability, and authority for empowering people has been communicated and delegated down to the individual manager or supervisor.

The first time the survey is administered, a baseline can be established. Subsequent measurements will indicate the extent to which a manager's empowerment behaviors (his or her paradigm) are moving (shifting) in the direction of an effective TQM paradigm (i.e., shifting from a 1, 2 or 3 to a 4, 5 or 6 on the response scale). The results from individual managers can be compared to their own past measurement periods to determine if the paradigm shift is in a positive direction. The composite mean scores from all managers whose empowerment behaviors are being baselined provide another valuable benchmark on which to assess individual or organizational progress.

The Empowerment Indicator Survey results may indicate the extent to which an individual supervisor or manager is empowering his/her people; however, the reality is that the score may also indicate the extent to which a supervisor or manager has been empowered by their own manager. The responsibility, accountability, and authority for empowering the work force should be first modeled and then delegated top-down in an organization. Subsequently, there should be a correlation between a supervisor's empowerment behaviors and his/her manager's empowerment behaviors. To reflect both possibilities - that a score has the potential of being owned by both supervisor and manager, an "NA" (No Authority; The responsibility and authority has not been delegated down to my manager) on the response scale is scored as a zero (0).

Results from the Empowerment Indicator Survey are intended to serve as a tool/catalyst for open discussion among all levels of management and the work force. Therefore, individual supervisors and managers sit down with their people, share the results and seek advise about how they can continue to manifest this paradigm-shifting experience.

Results from tracking manager's empowerment behaviors for the past year at the C-17 MDTS-Norman site have yielded the following preliminary findings and conclusions: 1) Program mean scores increased each measurement period indicating an increase in the level of empowerment experienced by the work force. Conclusion: Establishing a structure of management accountability that includes Empowerment standards of performance for empowering the work force, and a measurement system to provide managers periodic feedback relative to the direction and strength of their paradigm shift, will increase the level of empowerment experienced by the work force. It is beyond the scope of this analysis to conclude whether the reasons for increased empowerment scores were due to avoidance of low scores, increased awareness, or some other factors (e.g., Hawthorn effect). 2) Senior managers (rated by their direct reports - the middle managers) received the highest ratings (Mean = 4.74) of all managers or supervisors rated. Conclusion: Middle managers believe they are being empowered by senior managers and the paradigms of senior managers are shifting in a positive direction. 3) Middle managers (rated by first-line supervisors and their work force) received the lowest ratings (Mean = 3.79) of all managers or supervisors rated. Conclusion: While middle managers felt empowered by senior managers, first-line supervisors and the work force, in general, did not experience or enjoy a comparable level of empowerment. Middle managers, and to an extent first-line supervisors, appear to be a major inhibitor to the releasing/delegation of empowerment. 4) The items which were rated the lowest across all management levels were within the Maintain the Self Esteem subscale (particularly items 20 and 26). Conclusion: Management training in this area is recommended. 5) Overall empowerment mean scores for the organization increased when quality-productivity measurement systems were employed by the work force. Conclusion: Quality-productivity measurement systems appear to be an effective tool and catalyst which facilitates increased work force involvement in the constant improvement of processes and products. There are, however, other factors that could explain this increase.

CONCLUSION

The progress of a quality program is measured in years rather than months. Much of the progress achieved in the past eighteen months at the MDTS-Norman site is centered around quality awareness, establishing a quality infrastructure, measurement systems, and new skills. The effective implementation of empowerment, and Total Quality management as well, will include a model for management accountability and will not neglect a paradigm-shifting set of experiences for the managers involved in training - with periodic self-assessment and retraining. Unfortunately, far too much of TQM training is not addressing this critical cultural change and are, instead, focusing resources on quick fix techniques and image building which only yields the veneer of success. The reality (i.e., the experience of the work force) is change that does not substantively change anything nor does it make anything better - only different.

EMPOWERMENT INDICATOR SURVEY
(Manager's Form)

__Directions:__

After reading each question, use the following response scale and circle the number on the Answer Sheet that accurately describes the degree to which the following statements reflect your personal experience.

__RESPONSE SCALE__

1. Practically none; to a very small degree
2. Not very much; to a small degree
3. Moderately (on the low side)
4. Moderately (on the high side)
5. Very much; To a high degree
6. Extremely; To a very high degree

NA No Authority: The responsibility and authority has not been delegated down to my manager

__Sample:__

	Low					High	
11. The degree to which your manager provides opportunities (being asked) to share your ideas.	1	2	3	4	5	6	NA

EMPOWERMENT INDICATOR SURVEY
(Manager's Form)

1. The degree to which your manager provides opportunities which facilitate within you feelings that your job belongs to you -vs- belongs to the company.
2. The degree to which your manager's actions, decisions, and communications foster a working environment which provides opportunities for you to make things better and better.
3. The degree to which, time permitting, your manager provides opportunities which allow you to tackle problems normally not your job.
4. The degree to which your manager provides opportunities for taking on the challenges that affect your performance (-vs- not getting the opportunity or having the responsibility taken away).
5. The degree to which your manager provides opportunities which facilitate the understanding that your job really counts for something (-vs- doesn't really matter).
6. The degree to which your manager provides opportunities for you to discuss anything related to your job that is really important.
7. The degree to which you are asked by your manager to help solve problems or your opinions are sought.
8. The degree to which your manager provides opportunities to solve your own problems -vs- having someone else solve your problems for you (i.e., the degree to which you have ownership of solutions that affect your performance).
9. The degree to which your manager, or the system, lets you know how well you are doing (i.e., receive prompt, regular and meaningful feedback concerning your performance).
10. The degree to which your manager's actions, decisions, and communications foster a working environment in which your teammates can be trusted.
11. The degree to which your manager provides opportunities for you (being asked) to share your ideas.
12. The degree to which your ideas, if deserving, receive credit or recognition.
13. The degree to which your manager provides opportunities for you to have some say in how things get done that affect you or your performance (-vs- someone else making those decisions for you).

14. The degree to which you feel your ideas, opinions, or contributions are listened to by your manager.

15. The degree to which your manager has encouraged your team to develop and manage your own quality-schedule performance/feedback system (-vs- having someone else managing the feedback system).

16. The degree to which your manager's actions, decisions, and communications foster a working environment which enhances and reinforces the concept that your job is a part of who you are.

17. The degree to which your manager makes available adequate resources to do your job.

18. The degree to which information essential to your job/performance is shared by your manager.

19. The degree to which your manager's actions, decisions, and communications provide opportunities for you, and your job, to perform an important role on your team (-vs- feeling like you or your job doesn't count).

20. The degree to which your manager says something constructive about your performance daily.

21. The degree to which you feel your manager is providing adequate direction.

22. The degree to which your manager helps everyone on your team understand how their individual or team's mission fits with the overall mission of the program.

23. The degree to which adequate support from your manager is provided to do your job without your manager taking that support or responsibility from you.

24. The degree to which your manager provides opportunities which give you the feeling that you are on the inside (e.g., shares information vital to the performance of your job, invites you to work some issues important to the team/organization).

25. The degree to which your manager provides opportunities for you to demonstrate responsibility, accountability, and authority (-vs- just doing whatever you are told).

26. The degree to which your manager shares your team's successes with the rest of the organization.

27. The degree to which your manager makes your team aware of how well they are doing.

28. The degree to which your manager's actions, decisions, and communications facilitate the belief that your job is important and therefore, it is easy for you to connect your job to the organization's common purpose/mission.

29. The degree to which your manager provides opportunities which facilitate the belief that you can make a difference.

30. The degree to which your manager's actions, decisions, and communications foster a working environment which facilitate the desire among your teammates to really make things better and better.

31. The degree to which your manager has provided people on your team opportunities, responsibility, accountability, and the authority for taking on the challenges that affect your team's performance (-vs- having opportunities, responsibility, accountability, and the authority taken away).

32. The degree to which your manager has helped you understand how your individual and team's mission fits with the overall mission of the program/organization.

EMPOWERMENT SUMMARY

SUBSCALE	ITEM NO.	FUNCTIONAL AREA SCORE	PROJECT'S TEAM SCORE	GAP
MAINTAINS SELF ESTEEM	1			
	5			
	9			
	16			
	20			
	22			
	26			
	27			
	28			
	29			
	32			
SUBSCALE SCORE				
LISTENS AND RESPONDS WITH EMPATHY	2			
	6			
	10			
	14			
	17			
	30			
SUBSCALE SCORE				

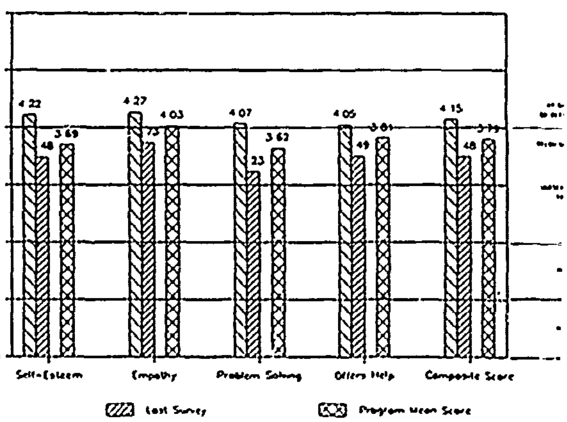
SUBSCALE	ITEM NO.	FUNCTIONAL AREA SCORE	PROJECT'S TEAM SCORE	GAP
ASKS FOR HELP IN SOLVING PROBLEMS	3			
	7			
	11			
	13			
	18			
	24			
SUBSCALE SCORE				
OFFERS HELP WITHOUT TAKING RESPONSIBILITY	4			
	8			
	12			
	15			
	19			
	21			
	23			
	25			
	31			
SUBSCALE SCORE				
EMPOWERMENT SCORE				

ZAPP!!! Empowerment Survey Results

Subscale	Question	Mean	Minimum	Maximum	Std Dev	Varianc
Maintain Self Esteem	1	3.67	2.00	5.00	1.24722	1.5555
	5	4.00	3.00	5.00	0.81650	0.6666
	9	3.33	3.00	4.00	0.47140	0.2222
	16	4.00	3.00	5.00	0.81650	0.6666
	20	2.67	1.00	4.00	1.24722	1.5555
	22	3.00	2.00	4.00	0.81650	0.6666
	26	4.33	4.00	5.00	0.47140	0.2222
	27	4.00	3.00	5.00	0.81650	0.6666
	28	4.33	4.00	5.00	0.47140	0.2222
	29	4.67	4.00	5.00	0.47140	0.2222
	32	4.67	3.00	6.00	1.24722	1.5555
Self-Esteem	3.88					
Listens and Responds with Empathy	2	5.33	5.00	6.00	0.47140	0.2222
	6	5.00	3.00	6.00	1.41421	2.0000
	10	2.67	1.00	4.00	1.24722	1.5555
	14	3.00	1.00	5.00	1.63299	2.6666
	17	4.33	4.00	5.00	0.47140	0.2222
	30	4.00	3.00	5.00	0.81650	0.6666
Empathy	4.06					
Asks for help in Solving Problems	3	3.00	2.00	4.00	0.81650	0.6666
	7	3.67	3.00	4.00	0.47140	0.2222
	11	3.67	3.00	4.00	0.47140	0.2222
	13	3.33	3.00	4.00	0.47140	0.2222
	18	3.33	2.00	4.00	0.94281	0.8888
	24	3.67	3.00	4.00	0.47140	0.2222
Problem Solving	3.44					
Offers Help Without taking Responsibility	4	4.00	3.00	5.00	0.81650	0.6666
	8	4.00	3.00	5.00	0.81650	0.6666
	12	3.67	3.00	4.00	0.47140	0.2222
	15	4.00	3.00	5.00	0.81650	0.6666
	19	3.33	2.00	4.00	0.94281	0.8888
	21	4.00	3.00	5.00	0.81650	0.6666
	23	3.67	1.00	5.00	1.88562	3.5555
	25	4.00	3.00	5.00	0.81650	0.6666
	31	4.00	3.00	5.00	0.81650	0.6666
Offers Help	3.85					

Self-Esteem	3.88	Jul-9
Empathy	4.06	
Problem Solving	3.44	
Offers Help	3.85	
Composite Score	3.81	Per cen Respons 42.8

ZAPP!!! Alert Zones



REFERENCES

[1] Adams, Kenneth L. and Kinchen, Kathy, "Successful Implementation of TQM," Published by Change Navigators, Inc., P.O Box 1118, Loveland, Colorado 80539.

[2] "A Checklist of Qualities That Make a Good Boss," Nation's Business, November 1984, p. 100.

[3] Covey, Stephen R., The 7 Habits of highly Effective People, Simon & Schuster, Inc. New York, 1989.

[4] Byham, William C., Zapp! The Lightning of Empowerment, Development Dimensions International Press, Pittsburgh, 1989.

[5] Herzberg, Frederick, "One more time: How do you motivate employees?" Harvard Business Review, September-October 1987, No. 87507.

[6] Kilmann, Ralph H., Beyond the Quick Fix, Jossey-Bass Publishers, San Francisco, 1988.

[7] Juran, J.M., "Made in USA - A Quality Resurgence," The Journal for Quality and Participation. March 1991, pp. 6-8.

[8] McGregor, Douglas, The Human Side of Enterprise, McGraw-Hill Book Company, Inc., New York, 1960, p. 6.

[9] Bram, James, "Gallup Poll shows CEOs passing Quality Buck." Quality Update, Philip Crosby Associates, Fall 1990, Winter Park, Florida.

[10] Roth, William, "Why aren't Leaders Leading?" The Journal for Quality and Participation. June 1991, p. 76-80.

ABOUT THE AUTHOR

Dr. Graydon Dawson is the Courseware Product Assurance manager for Simms Industries Inc. - a subcontractor to McDonnell Douglas Training Systems' C-17 Aircrew Training System in Norman, Oklahoma. Simms Industries provides scientific, engineering, training systems design and development, and management services to the Federal Government (DOD and DOE) and commercial enterprises. Graydon has spent 22 years in the field of education/training and development. Before initiating the quality improvement activities at McDonnell Douglas, Graydon served as a performance analyst and instructional systems design specialist conducting organizational and human performance studies for Electronic Data Systems Corporation (EDS), AT&T and General Motors Corporation. Graydon has also served as a full-time professor of Instructional Systems Design, Interactive Technologies and/or Division Chair for Texas Christian University, Oklahoma State University, Friends University and Southern Nazarene University (SNU). Besides working with McDonnell Douglas, Graydon is currently an adjunct professor for SNU teaching courses in Business Research/Applied Statistics, Principles of Management, and Systems Management for the Management of Human Resources (MHR) program. Graydon has presented over a dozen technical or research papers at national conventions and has published over 35 articles - the latest in The Journal for Quality and Participation (December 1991).

THE CHALLENGE OF DEVELOPING A COMPLEX TRAINING SYSTEM WITH AN INTERNATIONAL TEAM

Hans-Peter Engel
Program Manager 'Training Concepts'
Wegmann & Co. GmbH
Kassel, Germany

Greg Swick
Program Manager 'Application Software and CIG'
Advanced Simulation
Bolt Beranek and Newman Inc.
Cambridge, MA 02138

ABSTRACT

The future improvements in our industrial training technology base will include 'lessons learned' from our global allies This paper will describe the process and iterative engineering prototype methodology that was used to achieve an international training system success in a short time.

The recent combat training simulator program for the field training of armor personnel (Ausbildungsgeraet Gefictsimulator Panzertruppe) or AGPT represents the development, test and production of a platoon tactics trainer for use as part of an integrated training approach for the German Panzer Corps. The AGPT program also represents the cooperative effort of three German government agencies and three German and US contractors led by Wegmann GmbH, who formed a working group and proceeded to build a successful tactical training system in a little more than one year.

On December 21st of 1990, the German Army tanker school of Muenster reported a successful Troop Trial of the AGPT platoon set at Dornstadt, Federal Republic of Germany. In approximately 13 months from contract award to troop trials, a contractor team consisting of three companies on two continents, five separate engineering organizations in five worldwide locations (Boston, MA; Seattle, Washington, Woodland Hills, California, and in northern and southern Germany) — and all speaking two languages had designed a prototype trainer for evaluation.

The key to this successful and rapid schedule was a government/industry working group which traveled to the locations where the activity was centered in Germany, addressed the problems at hand in a spirit of cooperation, made immediate decisions, and then moved on to the next problem set.

A detailed approach to addressing international team program management issues and the resulting solutions will be presented in this paper.

INTRODUCTION

With defense budgets shrinking worldwide, the need for international cooperation in simulation and training has never been more acute. This trend is compounded by the need, especially in Europe, to reduce the environmental impact of maneuver training. No better example of a solution to these two problems exists than the recently successful AGPT program.

The AGPT, (Ausbildungsgerät Gefechtssimulator für die Panzertruppe), is a team tactics trainer at the platoon level for the Leo 2 Main Battle tank. It was built for the Bundeswehr by an international contractor team and will reduce the environmental impact of Leo 2 training by using the most recent simulation technology. The AGPT is also an example of using US developed technology implemented by a German led contractor team to meet the system requirements of a NATO military customer.

The AGPT requirement was developed by the Bundeswehr and issued for competitive procurement in December 1988. The requirement was to develop a turnkey simulation system which would replace the field training now used in all NATO countries. Key requirements envisioned and realized to accomplish this training were;

- a high detailed and culturally correct German database;
- a Semi-Automated Force (opposing and support);
- a scenario based instructor station that would allow a non-computer literate instructor to control all training; and
- a medium fidelity crew environment with image generator for out-the-window and thermal training

Economic realities were apparent in setting the requirements The government developed the strategy of large numbers of the platoon training sites and stated that the entire system must be self-contained. The total infrastructure planned was a concrete slab and appropriate electrical power. The system contractor had to develop, deploy and support the turn-key systems. The government supplied only the electrical power and instructor power to the sites.

To meet these requirements Wegmann assembled a German/US contractor team. Using Wegmann's knowledge as the Leo 2 turret developer and a builder of Leo 1 simulators, they added to the team US technology developed under the Darpa SIMNET tactical team trainer project. Wegmann established a team that spanned thousands of miles from Woodland Hills, California and Seattle, Washington thru Cambridge, Mass and then combined that with Wegmann's own facilities in northern Germany, Kassel and southern Germany, Fürstenfeldbruck.

The program team started development early, but the contract was awarded in 11/89. The real time Computer Image Generator (CIG) demonstration occurred the same month in Seattle. Intensive development proceeded until the integration of the prototype in Cambridge in 9/90 and the successful troop trials concluded at the Rommelkaserne, Dornstadt, Germany 12/29/90. The program is now in production preparation with the final of 15 production platoons due to the Bundeswehr in 3/93.

LESSONS LEARNED SHOW FOUR KEY FACTORS TO SUCCESS

#1 Industry Working Group

The first factor is the establishment of a government/ Industry Working Group. This term was very descriptive, because the group saw as its goal to do the work of the project, not to discuss merits of this proposal or that The group consisted of the key members of the government; the program office, technical management, the training school and tactics development; and the key program members from the contractor team; program and technical The group traveled to the point where the action was (e g troop trials) and made decisions on the spot. (In 13 months the government personnel spent 12 weeks in the US and approximately 50% of their time on travel.)

Decisions of the Working Group were recorded in the Detailed Specification and the protocols of meetings The Detailed Specification mapped to the basic requirement, the Lastenheft, on a paragraph by paragraph basis and served to amplify and explain the requirements (No requirements allocation was done here, this was done in the traditional manner in the Software Requirements Specification (SRS's) and the module specifications to allow testability of software and hardware components before integration.) The Detailed Spec paragraph for each requirement was discussed and negotiated by the Working Group. In general, as paragraphs were agreed upon the members of the working group would sign up to them

The group was empowered to make decisions and they made them. (It is important to note that senior management from the government or industry was very seldom present, except at social occasions and to lend support, and never was a decision of the working group overturned by senior management of the government or the contractors.) Decisions were almost always made on the spot with the information available, knowing that delaying for more complete information was not really possible. Complete information is never available and risk is never fully reduced, so as long as progress could be shown, work continued. If future events showed that the decision was wrong, it was changed No score was kept, and everyone moved on.

The government made decisions on implementation and the contractor on the technical underpinnings If the contractor could show that the design was supportable within the maintenance concept for fielding, he could continue. This was a fixed price contract with all penalties for late delivery or non-performance on the contractor, so all technical decisions were those of the contractor.

It is also important to note, that all parties knew there would be no change order activity. The team did a 'give and take' to get the best possible training system, knowing that everyone had to succeed in a zero sum game

#2 Rapid Prototyping Methodology

The second key factor leading to success was the use of a rapid prototyping methodology. This was made possible because the basic Distributed Simulation technology was proven. Wegmann divided the system amongst the team into logical parts based upon their experience in previous training systems. The interfaces were clean and there was little cross coupling.

The entire system was built quickly in an iterative fashion. A small section of the database was built and reviewed by the military specialists. Meanwhile, it was run on all of the target machines in the distributed environment to ensure compatibility. The User Interface was prototyped and reviewed by the trainers from the Armor School who would have to use it. As they saw the user interface they could imagine the functionality of the Semi-Automated Forces and the missions needed for training; and provided valuable feedback. Discussion was limited and usually took place around a barebones piece of equipment, not a table full of abstract concepts.

Perceptronics, Inc. built mock-ups by hand, which served to give the user a feel for the environment of the crew cabin and aided in integration and tool debug. BBN built a system backbone in Ada quickly and then used that backbone to call large portions of software in 'C' developed in previous simulations. By doing this, although the functionality was not correct and the software not supportable, the User got early exposure to the system and directed its implementation based upon use of this rapid prototype model. As the user requirement was solid BBN would then design the modules to meet the understood requirement in Ada. This also served to eliminate development risk as the technical team gained insight into how well the Ada/C prototype ran they could estimate how the full Ada run-time environment would operate.

The feedback became the requirements analysis and trade analysis which became the basis for system design. As the engineers listened to the User's inputs the design evolved.

#3 Heterogeneous Technical Mangement

The third factor in success was the technical management. Wegmann chose a heterogeneous team which had strengths which complemented one another. There was

also a strong tendency to use the experience already gained in building Leo2, Leo simulators and in building SIMNET As the main contractor Wegmann pushed the adaptation of Commercial and developed technology to the AGPT requirement. This 'reuse of technology' approach led to a system which evolved quickly as the technical specialists all understood their basic technology and were allowed to implement it in a way familiar to them Large pieces of commercial off-the-shelf software were used. The motto was use and adapt what exists to meet the requirement at hand.

#4 Interpersonal Bonding Within Team

Finally, there developed strong interpersonal bonds which helped the through rough times. People from different companies and across the government/industry boundaries developed a respect for one anothers' capabilities, integrity and competence. The government personnel trusted that the Wegmann team was doing their best to deliver a good training system. When discrepancies were noted there was the assumption that they would be fixed. These were status checked, however, the assumption was based upon trust.

Cultural problems and understandings existed, of course. But the attitude of the people working on the project was a fresh one and people respected the cultural diversity as they did the functional and technological diversity. The heterogeneous team - culturally and functionally - never felt threatened by one another, and seriously wanted to learn and enjoy the others culture. This took place at work and in social settings. The team enjoyed these interactions, (late at night with pizza or in a formal dinner), learning about the culture of the other part of the team also meant learning something of his requirements.

SUMMARY

In conclusion, by all measures the AGPT is so far a successful program. It is on schedule and at no additional cost to the government. The initial users give it high ma[illegible]. In the end AGPT is the result of the energy, attitude and work of a relatively small group of dedicated people. These are people who are not afraid to take a risk, who will delegate a task and a responsibility, and who are not afraid to listen and act on an idea to turn it into a reality.

ABOUT THE AUTHORS

Peter Engel has worked for Wegmann GmbH since 1986. He has the overall responsibility for training concepts prior to start of developing a training system. A member of the AGPT team from the very beginning he was responsible for the initial inputs of the rapid prototyping process of the crew cabins as well as for the User interfaces in the instructor console. Before joining Wegmann, Mr. Engel served in the German Armor corps rising to the rank of a company commander. Currently he is XO in a reserve tank battalion.

Greg Swick is currently Program Manager in the Advanced Simulation Business Unit of BBN Systems and Technology Division. As such, Mr. Swick exercises overall authority for program schedule, cost, and work scope for BBN's portion of the contract.

Previously Mr. Swick worked in the defense electronics business in DoD and international military procurements. Mr. Swick flew a F-4 in the Air Force and was a simulator instructor.

Streamlined Source Selection
or
Write Your Own Spec!

by A. Edward Dietz
AAI Corporation
Hunt Valley, Maryland

ABSTRACT

Recently several agencies have applied Total Quality Management (TQM) principles to new contract starts and requests for proposals (RFP). One form of solicitation is called Streamlined Source Selection. This paper explores the costs, benefits and problems associated with the Streamlined Source Selection approach from a contractor point of view. Is the source selection process improved? Is the feedback from industry really used by the agencies? Is the government evaluation process more equitable, easier, better or cheaper with this streamlined approach? Are the total pre-award costs reduced, increased or merely shifted from one place to another? Are contracts awarded more quickly? Does the specification better represent what is needed or what will be produced or both? Is the resulting contract of higher quality? Does the government get a better bargain? Are contracts completed more quickly or with fewer changes or fewer cost problems? In short, are the expected TQM benefits obtained?

INTRODUCTION

The streamlined source selection approach generates several RFP drafts with extensive consultation between the contracting agency and industry to improve and expand the material between drafts. The formal RFP is then issued with a relatively short response time. One feature of this streamlined approach is that each bidder writes many of the pertinent contract documents as a part of the proposal. These documents usually include the specification, the system engineering master schedule, portions of the statement of work, a software development plan and other program plans and data items. Once the proposal is submitted the government contracting agency evaluates all proposals on an accelerated schedule.

Much of this paper will be focused on the technical specification generation process and its problems and benefits. The discussion and questions apply equally across the other tailored documents such as the schedule, statement of work, logistical plans etc.

DRAFT RFP

The draft Request for Proposal (RFP) process for streamlined source selection starts out much like the traditional draft. A set of preliminary documents is issued, perhaps with one or two major segments missing. Contractors are asked to comment and make suggestions in writing. But there are differences. The documents usually include only a very general top level technical requirement. The schedule may have only an end date and a set of intermediate milestones without dates. The contractor is expected and required to eventually fill the technical, management and contractual gaps. Some of these gaps can be filled through general suggestions during the draft process. Others can be handled differently by each contractor when he eventually submits his own proposal.

The contacting agency pays a good deal of attention to the suggestions of bidders. Many areas are revised and expanded based directly on contractors suggestions. A dialogue occurs in both classic bidder's conference form and single contractor face to face discussions.

Additional drafts are generated to refine the process. Missing segments are added. Appropriate contractor suggestions are incorporated. This process often continues over a number of draft RFP's. Contractors are encouraged to write, though not submit, portions of their proposals during this period.

FORMAL RFP

Once the formal RFP is released, a serious effort is made to compress the time remaining before award. Thirty day response times are common for proposals valued at tens of millions of dollars. Even shorter response times are required for such items as past performance data. The contracting agency gives itself a tight and meaningful time limit for evaluation as well.

The formal RFP is the last point for additions, deletions and changes in requirements. Many of these changes come from the discussions and comments mentioned earlier in this paper. Others may come from last minute changes within the government. The changes may run the gamut of technical, logistics, management and contractual.

While no one enjoys change for its own sake, all can recognize that it will occur. However, one type of change is particularly difficult in the streamlined source selection atmosphere. When the government changes the volume of response or the nature of the response only at the final RFP, the bidders are left with very little time to respond. The addition of major data items or questionnaires has added 10% to the effort required during the 30 day response period.

PROPOSAL PREPARATION

Various approaches have been applied to the actual material required from the bidders. Some procurement efforts still require a classic expository description of how the job will be performed. Others focus on particular areas of interest with very specific and stringent page limits. Yet others drastically limit the amount of general text and rely on the contractual documents to define and describe the task.

Now what are "contractual documents" in this situation? They are familiar items such as statements of work, specifications, logistics plans and major milestone schedules.

The government approach to contractor generated schedules is particularly innovative. Not only must dates be proposed but the success criteria for that milestone also must be proposed. Both the dates and the criteria can and do vary from one bidder to another. These milestones are major items such as CDR, hardware-software integration start, factory acceptance test start, ship to installation site etc.

Likewise the specifications can vary. The bidders are free to set numerical limits that are practical, applicable to their approach, and relevant to the overall procurement goal. Of course this freedom is constrained by the general technical limits set by the RFP. However, this freedom is frequently quite broad. The contractor is allowed more innovation than in past procurements. Besides choosing an approach, the contractor can specify intermediate requirements and measurements. He must convince the government that these values are plausible and consistent with achieving the required result.

PROPOSAL EVALUATION

In this phase of activity the contractor insights are limited. Agencies continue to closely control the details of this process in the interest of fair play. That said, new situations occur with the streamlined approach that appear different from the standard evaluation concerns.

Under the streamlined system each bidder submits his own specification and detailed schedule. Variations are allowed in the technical requirements and in the way they are implemented. Such schedule items as program phases and testing times are allowed to vary among bidders. The general schedule and the top level specification are the same for all. Given this starting point, can the government evaluators reconcile several varying requirements and schedules?

Consider two bidders X and Y who prepare proposals for a new trainer. Suppose bidder X proposes a long development schedule with a CDR later than other bidders. Bidder Y proposes a shorter development schedule with a CDR at the 12 month mark. Does this mean that Y understands the task better or has more experience? Perhaps, but read on. Now suppose X also proposes a short fabrication and coding cycle compared to Y. Does this mean that X has taken advantage of the longer development time to thoroughly plan the job and minimize coding effort? Or has X missed the mark entirely and simply underestimated the coding and debugging effort?

Other questions which occur are:

- How is it possible to uniformly and fairly evaluate a number of bids where the specifications are different?

- Is there a way to correlate between a permissive (weak) specification and a lower price? If so, is this combination perceived as a desirable, awardable situation? Or is this a cause for discussions to tighten the specification?

- What about the case of a tight specification and a low price? Is this a buyer's delight or a cause for suspicion about cost realism, technical understanding and completion risk?

- Is an aggressive schedule and low price a plus or merely an added risk?

We are aware that the agencies apply standards of acceptability uniformly to all proposals. We suspect that this job is more complicated and difficult with the wide variations amongst bidders and the contractor-created requirements allowed by the streamlined approach. From the awards made to date, we can see no pattern concerning an optimum approach to be used by bidders.

On the positive side, many contracts have been evaluated and awarded with commendable speed. Does this speed stem from a smooth evaluation process or from the extensive contractor and government improvement efforts during the draft RFP phase?

We encourage a future paper exploring the evaluation situation from the government viewpoint.

POST AWARD ACTIVITY

The inter-service/industry team is just embarking on this phase for contracts that were awarded by streamlined methods. In theory the team efforts that occurred during source selection should have several benefits:

- Adversarial relationships ought to be reduced as both parties have made a strong contribution to the effort before the contract is signed.

- The contract documents should be familiar to both parties at the start. Thus the learning curve to merely understand what is required should be shorter.

- There should be fewer problems or disagreements about the specification since the much of the specification should reflect the bidder's technical approach.

CONTRACTOR SPECIFICATION WRITING BENEFITS

There are at least two interpretations of the "Write Your Own Spec!" title phrase. One response, from many engineers, is joy. Many engineers have spent years responding to requirements written by others. Those requirements often <u>seem</u>:

-Inconsistent.

-Vague.

-Too specific.

-Unrelated to the task at hand.

-Narrowly drawn without considering the whole problem.

-Written around someone else's design.

-Outmoded.

-Too visionary.

-Impractical.

It is no accident that the preceding list is contradictory on its face. Note the emphasis on the word "seem." Make the utopian assumption that the customer organization has a perfect understanding of its need. Then that need must be translated into a written specification. Then the bidder's technical staff has to understand the written word. No wonder the engineer in industry is frequently in the position of not fully understanding what the customer's requirement may be. Frustration and human nature then operate to produce a reaction of "I could write a better spec than this blindfolded" or something similar. Now present that same individual with a real chance to write a specification for his or her area of expertise. Frequently the initial reaction is open mouthed delight.

After a while much of the happiness fades. When the task is contemplated in detail, it is soon obvious that the contractor's engineers have bitten into a large mouthful of additional work and worry. Now the contractor must fully understand the requirement, make a number of painful choices using achievable numbers, and produce a result that he and his corporate associates must live with as successful bidders. By the way the result must still satisfy the wants and needs of the customer. A few trips around this loop of thinking can produce a sarcastic reaction of wishing that the specification writing job were back with the customer. Hence the alternate interpretation of "Write Your Own Spec!"

Having made the contrasting points above, a little balance is needed. Contractor engineers can indeed write specifications. They do so constantly for their own suppliers and as detailed supplements to government documents. The question is whether or not a change in the prime contract specification writing responsibility is beneficial. The answer is not clear at the moment as the streamlined approach is new. Nevertheless some measures of success suggest themselves for use during contract performance.

- Was the design review process smoother with fewer resubmissions of documents required?

- Were there fewer problems in generating second tier specifications during the job? This could be judged from the number of government complaints generated about the submittal of appropriate data items.

- At the end of the contract, were there more or fewer waivers, deviations, and specification change notices compared to past experience with government generated specifications? Obviously this answer must be adjusted to allow for engineering change proposals submitted for reasons unrelated to specification problems.

- Were there fewer test discrepancies in general? Were there fewer discrepancies that were resolved by specification changes?

- Was Initial Operational Test and Evaluation more satisfactory?

COST CONSIDERATIONS

The bidders get to write their own individually tailored contract requirements. This gives them some degree of unique advantage in the bidding and, if successful, in the performance of that contract. But each bidder is doing this writing at his or her own bid and proposal expense. This expense is eventually borne by the government as part of reimbursable General and Administrative costs. If the government wrote these documents there would be only a single effort instead of many. But the documents would contain only the government point of view and would necessarily have some compromises to allow all potential bidders an equal opportunity. If the effort is merely moved from the government to the contractor has anything been saved? Or worse has the overall taxpayer cost been increased since several bidders each must write contractual documents. Or is this increase in bid costs trivial because the process creates a better training device at a lower contract price?

COMMENTS ON PROCEDURE

The discussion process during the draft RFP phase is a significant step forward. Serious consideration is clearly being given to contractors ideas. Many suggestions have been implemented.

Major changes to the proposal applied at formal RFP go against the grain of the process. The opportunity for consultation is lost at that point. Many reworked drafts mean little if the last formal one adds a lot. Efforts in advance of the formal RFP by the bidders may be wasted.

Most of the streamlined proposals have page count limitations which initially appear severe. But these proposals also have contractual documents, appendices, plans, supplements and initial data item submittals. None of

these items apply against the page count limit. The items all require effort by the contractor to write; they all require effort by the government to read. On a recent proposal that was formally limited to several hundred pages the actual page count submitted was over 5,000. What has been saved by the page limitation?

With or without page count limitations, the streamlined approach has not yet controlled the proliferation of data items required with proposals. The repetitive cost of this data item generation across numerous bidders and numerous proposals is large and growing. Although many of these items contain repetitive boiler plate, the cost of constantly revising and resubmitting them is not trivial. Further streamlining of the acquisition process could include submitting many plans on an annual or as revised basis.

Contracting agencies are encouraged to get other areas of the contracting process to join the trend. The logistics requirements on many jobs are certainly open to tailoring. So are data items in general.

USERS

One major interested party has been neglected so far in this discussion. What about the eventual training device user? Does the streamlined source selection process affect the interests of the organization that has the basic need?

In theory, if the process has resulted in a better set of contract description documents then the user has an increased chance of getting a superior device.

If the user is a part of the evaluation team, he or she must suffer through the same multiple 1evaluation concerns discussed elsewhere in this paper. The user is typically more familiar than the contracting agency with the actual parameters needed to do his job. Therefore the user may well have to spend more time evaluating different contractor generated specifications compared to the time spent evaluating responses to a standard specification. The up side of this time investment is that the user may be able to make a more meaningful choice amongst bidders than in the past. This has benefits to the government but at the expense of an uneven playing field for bidders.

IN CONCLUSION

The key points of the streamlined source selection process are:

- Extensive government and industry cooperation on

  the RFP content.

- Bidders write unique contract documents tailored to their individual approaches.

- Considerable latitude is allowed in responses by different bidders.

- Proposal response time is shorter than past practice.

- Government evaluation time is short.

We believe that the streamlined source selection process has already met several of its goals. It produces a better set of contractual documents is equal or less time than previous approaches. It increases the involvement of the winning bidder early in the process. Speed of contract award has been increased. Additional attention has been paid to industry feedback. The specification will more closely match what will be produced. Thus gains in quality have already been achieved.

Deming claims, "Improved quality decreases cost because it reduces rework, mistakes, delays, and snags."[1] We suspect that total proposal costs for both government and industry have remained equal at best and may have increased. Comparative measures of cost for both business and government are hard to identify but are the only way to resolve this question.

The issues of fairness and equality in competition are also difficult to measure. We suspect that there have been losses in equality due to the increased contractor initiatives allowed. This is not necessarily bad in the context of improved quality.

It is too early to evaluate whether contact performance has been improved by the process. We invite government response in these areas and in overall evaluation of benefits of this source selection approach.

About the Author

A. Edward Dietz is a Program Manager in the Training and Simulation Division of AAI Corporation. He was the program manager for the EF-111A Operational Flight Trainer and the A6/F14 Aircrew Training System. He has held a series of engineering and management positions on radar, electronic combat and flight trainers at AAI. He has been a proposal manager on numerous bids including several conducted under Streamlined Source Selection guidelines. He holds a B.S. degree in Electrical Engineering from Purdue University.

1.W. Edwards Deming, Out of the Crisis, 1986

CTASC-II TRAINING — KEEPING PACE WITH AN NDI ACQUISITION

by
George (Chip) S. Peratino, Jr.
Robert M. Cowell
EER Systems Corporation
Vienna, Virginia

ABSTRACT

This paper will discuss development and implementation of training systems that must be fully operational in an abbreviated development period due to the rapid deployment of a non-developmental item (NDI) system. The requirement is to develop the training curricula and associated training aids/devices in sufficient time to train individuals/crews to operate and maintain the operational systems as they are fielded. Specifically, this paper will address the training system developed for the Army's Corps/Theater ADP Service Center - Phase II (CTASC-II). This paper will explore how the schedule of the operational system impacts upon the development of the Training System and the associated training system cost and schedule.

Typical DoD system acquisitions span four to seven years from concept formulation to fielding of the system. NDI acquisitions can be accomplished in less than three years. Therefore, the training system concept, development, and implementation time for NDI systems is significantly reduced. While developing this training system to meet the objective of producing a trained crew to accompany the fielding of the first system in less than three years, decisions must be made regarding how and when to accelerate certain developmental steps, i.e., tailor the Systems Approach to Training (SAT). This paper will address which developmental steps can or cannot be omitted or accelerated and the results of doing so.

In discussing this developmental process, the paper will address the intrinsic training benefits that are available to training developers when using NDI equipment, such as immediate access to technical documentation and reduced acquisition cost of training devices. It will also discuss the various difficulties that are encountered when developing training systems for an NDI system; e.g., data rights that are proprietary to each hardware vendor.

NDI, in today's environment of shrinking DoD budgets, will be used to produce operational systems wherever possible. Therefore, it is essential that training developers learn how to be responsive in this type of environment. The training system development which was accomplished in support of CTASC-II training is one successful way.

INTRODUCTION

This paper will discuss training systems that are developed and implemented within an accelerated schedule to train individuals/crews to operate and maintain an operational system that uses non-developmental items (NDI). Specifically, the paper will address the training system developed for the Army's Corps/Theater ADP Service Center - Phase II (CTASC-II). The paper will explore how the schedule of the operational system impacts upon the development of the training system.

NDI OPERATIONAL SYSTEM

A non-developmental item (NDI) acquisition utilizes commercially available, off-the-shelf equipment that is used "as is" or modified to satisfy an operational requirement. An operational system may be composed of several NDI equipments which are integrated together with previously developed tactical equipment to perform specific functional requirements. NDI systems are being procured more frequently by DoD because of constrained budgets as well as increased mission emphasis to deliver systems in abbreviated acquisition schedules.

Use of NDI equipment has several intrinsic benefits. The NDI equipment is more economical to utilize because the development costs have been incurred by a commercial vendor who amortizes these costs over many different customers. Another benefit of an NDI acquisition is the availability of the equipment which significantly reduces the time it takes to develop the first unit. Procurement of the hardware can occur as quickly as one to three months. Typical Research and Development (R&D) DoD acquisitions can take from four (4) to ten (10) years from concept to delivery of the tactical system, whereas NDI acquisitions can occur within three (3) years. A third benefit of NDI acquisitions is product maturity. Often the NDI equipment has been commercially available for some time and has a reliable track record. However, because a specific reliability cannot be levied on NDI hardware vendors, operational

system designers utilize redundancy to achieve a higher system availability. Specifically, if 6MB of disk space is required, the system designers might utilize twelve 1MB disk drives to provide 100% spares or redundancy. If one of the disk drives fails, then another of the unused drives can be utilized immediately, therefore, system availability is maintained.

One role that is very prominent in an NDI acquisition is that of the System Integrator (SI). The SI is responsible for defining and assuring that all hardware/software interfaces are compatible. The SI also is responsible for developing system diagnostics and exercisers that test the system's performance. An expedient development method is to take advantage of the built-in diagnostics that each NDI equipment contains, wherever possible. The SI also writes interface software that utilizes existing commercially available operating systems and software. Furthermore, the SI is responsible for hardware interfaces such as equipment layout, electronic signal levels, and cabling. Finally, the SI is responsible for integration of the hardware and software components into a system.

NDI ACQUISITION STRATEGY

In an NDI acquisition, the formal developmental documentation of a typical acquisition is significantly reduced. Although most documentation required to get a program approved and funded, such as a Mission Element Needs Statement (MENS), is still required, often hardware specifications are not utilized but rather the system's required operational characteristics (ROC) and commercial specifications that are provided by the hardware manufacturer are substituted.

Furthermore, in an NDI acquisition, requirements contracts are often used for procurement of automated data processing equipment (ADPE), and the GSA Schedule can be utilized to procure other hardware components. Utilizing these existing contract vehicles significantly reduces the time required to procure this equipment and to deliver it to the SI. This procurement strategy also significantly reduces the time required to design and deliver the training devices. Once specified, acquisition of the training device can parallel and utilize the same contract vehicles as the operational system acquisition. In some cases the cost of the operational system is low enough to utilize an actual operational system as a training device.

Utilization of a government agency to perform the role of the SI will also reduce the amount of time required to begin and complete the integration because there is no formal procurement of contractual services. This procurement cycle can take from nine to eighteen months before contract award. Instead the acquisition agency can develop a statement of work for and send a work request to a government agency in a relatively short period of time.

THE CTASC-II SYSTEM

The NDI acquisition that we are going to discuss is the Corps/Theater Automated Data Processing (ADP) Service Center-Phase II (CTASC-II). CTASC-II is a mobile, survivable, ADP facility which is used by major subordinate commands at corps and theater levels. CTASC-II provides information processing support for logistical, personnel and medical Combat Service Support (CSS). The CTASC-II systems provide ADP service for Standard Army Management Information Systems (STAMIS) which include the Theater Army Medical Management Information System (TAMMIS) and the Standard Army Retail Supply System (SARSS). The US Army Program Manager (PM) for Tactical Management Information Systems (TACMIS) is responsible for the acquisition and life cycle support of CTASC-II. The Milestone 0 Major Automated Information System Review Committee (MAISRC) was conducted on 29 April 1987 and approval to commence proof of principal was given on 1 July 1987. The CTASC-II program successfully achieved a Milestone I/II decision on 28 July 1988 which approved the concept and the commencement of preproduction prove out. The initial CTASC-II, with a trained crew which had completed their ten weeks of crew training, was delivered to Fort Bragg, North Carolina, in March 1990 This system was used initially for the SARSS Software Acceptance Test (SAT) before going on line and then being shipped to Saudia Arabia for use during Desert Shield/Storm. As a result, the training course was developed and the training devices acquired within a five-month period so that the pilot crew training course could begin in September 1989. The Milestone III full production decision is anticipated in 2nd Quarter, Fiscal Year 1992.

The major components of the CTASC-II are the automated data processing equipment (ADPE) and the Communication Electronics (CE) equipment. These components are designed to operate with temperature, humidity, and dust levels being maintained by rigid wall shelters with Environmental Control Units (ECUs) to regulate temperature and humidity. These rigid wall shelters are mounted on and transported by Commercial Utility Cargo Vehicles (CUCVs). Both the ECUs and CUCVs exist in the Army inventory. The CTASC-II consists of three shelters with the majority of the ADPE in the central processing unit (CPU) shelter except for the operator interface equipment. The CE and operator interface ADPE are contained in the operations shelter. The third shelter is used for support equipment. Two ECUs are mounted on trailers that are pulled by the CUCVs. The typical CTASC-II field site layout is shown in Figure 1. The ADPE and CE are mounted in racks and the NDI equipment is highlighted in Figures 2 and 3. The redundant equipment is also shown.

The CTASC-II system is operated 24 hours per day by a crew of seven (7) Military Occupational Specialty (MOS) 74Ds. Three crew members work each 12-hour shift with

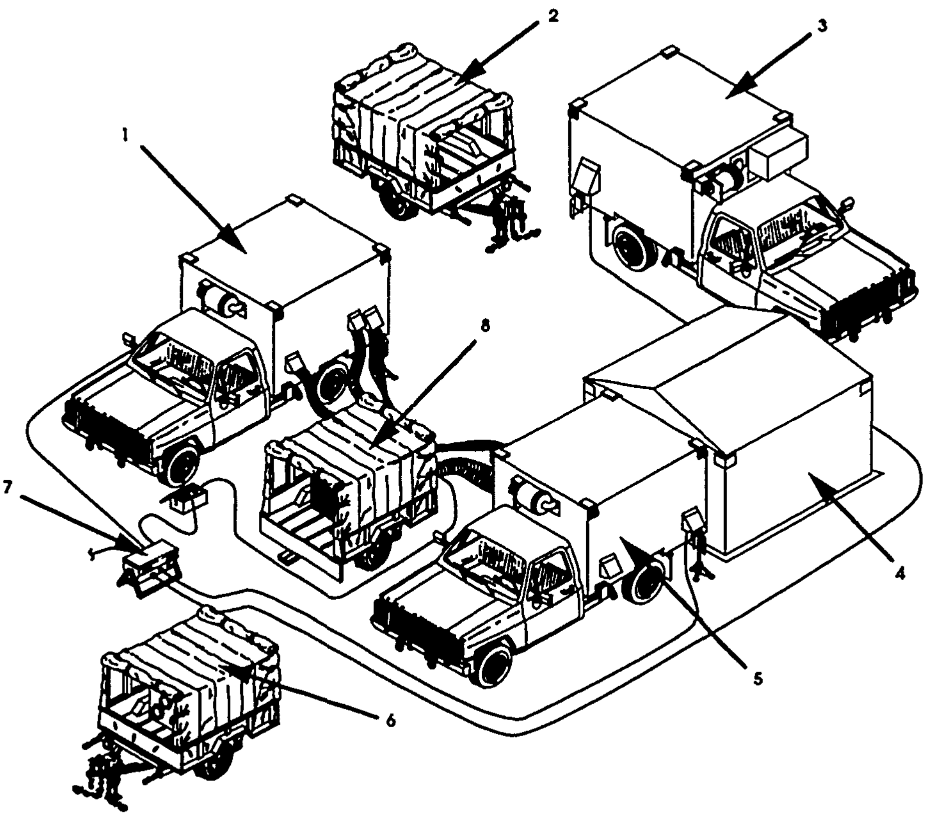


FIGURE 1: TYPICAL FIELD SITE LAYOUT

one additional floating member. The crew not only operates the CTASC-II system but performs preventive and limited organizational maintenance functions.

TRAINING SYSTEM DEVELOPMENT

Introduction

Training development for the CTASC-II system is based on the Systems Approach to Training (SAT). The SAT is a guide for training system development that is applicable to both formal and NDI acquisitions. The SAT process requires the following steps:

(a) Identification and analysis of the tasks performed in the duty position and the behavior required of the soldier.

(b) Design of training objectives.

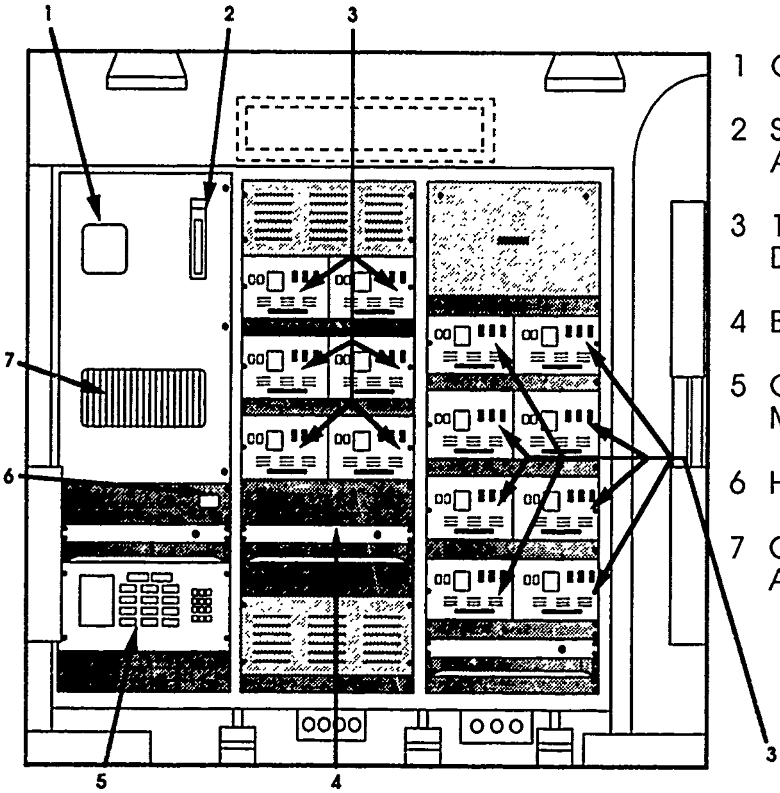
(c) Development of training programs and materials to achieve these objectives.

(d) Implementation or conduct of training.

(e) Evaluation of training graduates by criterion referenced testing.

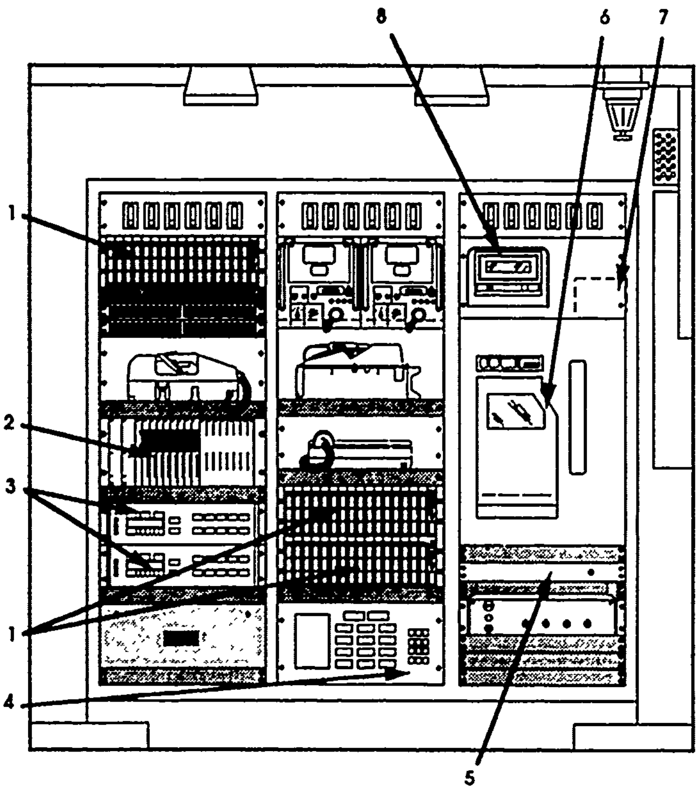
None of these five processes is done in isolation. The results of each process serve as inputs to one or more subsequent processes.

Implementation of the SAT process will normally take from one to three years. However, just as the NDI acquisition follows an abbreviated schedule, so does the training system development for that NDI system. An NDI training system development is similar in structure to a typical SAT development; however, certain development steps are either eliminated, consolidated, or tailored to achieve the required accelerated schedule. Training development for the CTASC-II system took a significantly lesser amount of time from its beginning in March 1989 to the date of the first class in September 1989. Shortening these steps was possible because senior 74D subject matter experts (SME) and EER contractor personnel with



1 CPU Card Cage
2 Streaming Tape Assembly
3 1.02 GigaByte Disk Drives
4 Blower Assembly
5 Omnimux 3200 Multiplexer
6 Hour Meter
7 CPU Circuit Card Assemblies

FIGURE 2: CPU SHELTER



1 Digitial Patch Panel Assemblies
2 V.22 Modems
3 V.29 Modems
4 Omnimux 3200 Multiplexer
5 Blower Assembly
6 9 Track Tape Drive
7 Mil-Std 188-114/RS 232 Interface Converter
8 8-MM Tape Archive Device

FIGURE 3: OPERATIONS SHELTER

many combined years of experience in the SAT process and specific knowledge of the CTASC-II system were brought together to complete detailed task breakdowns during the analysis phase. Much of the design work was accomplished during this phase which resulted in significant time savings in the subsequent design and development phases. In the following paragraphs, we will discuss the NDI training development steps and the tailoring of the normal SAT process.

Analysis

The purpose of the analysis phase was to inventory the tasks which must be performed by the 74D crew for the CTASC-II system in order to determine the skills and knowledges required to perform each task. The MOS 74D trainee characteristics and job standards as well as the maintenance concept for the CTASC-II system had the most significant impact on the formulation of these CTASC-II crew tasks and the subsequent design of the training program.

MOS 74D Trainee Characteristics and Job Standards. Operation and unit level maintenance of the CTASC-II system is one of the current job standards of the 74D, formerly a Computer Machine Operator. The job standards of today's 74D Information Systems Operator have changed from the requirements of the 74D Computer Machine Operator. The typical trainee obtaining the 74D Information Systems Operator MOS receives more training than the 74D of the past. The major duties of the former 74D Computer Machine Operator were to supervise or operate a computer system and/or auxiliary equipment in an automatic data processing environment in a fixed facility. Typical tasks that the 74D Computer Machine Operator was required to perform were all related to ADPE operations and/or administrative functions, such as mounting and dismounting magnetic tape and disk media or operating a computer console. However, the job standards of the 74D Information Systems Operator have significantly changed in today's information system environment. No longer is the 74D just an operator or administrator. Now he must also be familiar with troubleshooting and maintenance techniques of information processing systems equipment.

All personnel entering the 74D MOS training program will meet certain basic prerequisites. Each trainee will have a minimum of a high school diploma, be a U.S. citizen and possess or be able to obtain a secret security clearance. The physical qualifications of the 74D include being able to lift 50 to 70 pounds and having normal color vision. A 74D trainee must also have a qualifying score of 100 in the Basic Entry Aptitude area of technical skills.

Prior to attending the CTASC-II crew training course, the 74D has already attended a thirteen week MOS course conducted by the Computer Science School located at Fort Gordon, Georgia. This course provides the 74D with a basic introduction to computer concepts. He also receives training on input and output control and magnetic media of a data processing activity. This course also provides operating systems training such as IBM's Multiple Virtual Storage (MVS), UNIX and Job Control Language (JCL). Additional training is provided in DOS applications such as word processing, spreadsheets, and data base management systems. The 74D also receives basic knowledge training in the area of data communications. Since January of this year the 74D has begun to receive training on tactical computer systems.

Upon completion of his MOS training, the 74D Information Systems Operator has been trained to supervise, install, operate and perform unit level maintenance on multi-function/multi-user information process system(s), peripheral equipment and associated devices in mobile and fixed facilities. Additionally, the "new" 74D is responsible for connecting power generation equipment for operations and for operating environmental information processing systems. The 74D Information Systems Operator of today must also troubleshoot information processing systems to the degree required to fault isolate to the Line Replaceable Unit (LRU) and effect restoration of information processing systems by replacement of the LRU. The new 74D also operates and performs Preventive Maintenance Check Services (PMCS) on assigned vehicles and power generation equipment. The skills and knowledges that the 74D has acquired that relate to operation and unit level maintenance of the CTASC-II were included in the task inventory for the CTASC-II crew training course. During the design of the training program, a decision was made whether to select these tasks for additional training.

Maintenance Concept. The selected NDI maintenance concept has a significant bearing upon the development of the maintenance portion of the training system. In order to keep system availability at the highest level possible, remove and replace at the LRU level is the most practical maintenance concept. It allows rapid replacement of failed equipment so that the system can be returned to operational status. The CTASC-II interim maintenance concept, likewise, is remove and replace at the organizational level any LRUs that are not a hazard to the soldier. All other maintenance, to include depot, will be performed by a support contractor. The remove/replace maintenance concept significantly reduced the amount of functional tasks that the crew must perform. This, in turn, reduces the number of maintenance tasks that must be

taught and therefore reduces both curricula design and preparation as well as maintenance device requirements.

Design

After the analysis of the job is complete, the design of the training program is performed. Design involves the selection and conversion of tasks into objectives, the determination of test items, sequencing of information to be taught and the selection of the best media to support the transfer of knowledge and skills. The CTASC-II crew training course includes classroom instruction supplemented with over 60% hands-on practical exercise training. Twelve of the sixteen blocks require the trainee to successfully complete a written examination and a performance examination.

Task Selection Matrix. During the design process, the previously inventoried tasks were integrated into a task selection matrix. The task selection matrix is a complete listing of all tasks, individual and collective, that are necessary actions in the performance of a job or duty. A decision was then made as to which tasks would be selected as training items. All selected tasks were observable and measurable and had definable starting and ending points.

Usually the task selection matrix is developed using the Logistic Support Analysis (LSA) data. However, because that data was not available early enough in the CTASC-II Program, an alternative method was selected which would allow the training program to be developed and implemented in sufficient time for fielding. EER personnel and U.S. Army SME's formed a joint task selection board to develop the task listing. The individuals participating on this board were senior 74D SMEs and EER personnel with many combined years of experience in the SAT process and specific knowledge of the CTASC-II System. The typical design of a task selection matrix is a listing of the tasks that the trainee must perform. However, the task selection matrix that was developed for the CTASC-II was much more detailed. It included each individual step required to perform a task. An example of this format is as follows:

TASK: Perform disk maintenance.

Step 1: Format a disk with the disktest command.

Step 2 Test a disk using the disktest command.

Step 3: Partition a disk using the dsetup command.

TASK: Replace a 9-Track Tape.

Step 1. Unmount the 9-track tape for the tape drive.

Step 2: Obtain a new tape.

Step 3: Mount the new 9-track tape on the tape drive.

The training system developers also utilized the documentation available from the commercial vendors to perform the task analysis and develop the task selection matrix. The operational manuals, that generally accompany the NDI equipment, were used to identify those tasks to be performed on a specific piece of equipment. Each hardware vendor offered documentation regarding their hardware, some provided it free with their equipment, some for nominal cost, and some for average cost. Some of the hardware vendors also taught classes in the operation and maintenance of their equipment. Once the task selection matrix was developed, each task was verified by the training developers using the operational system located at the training site. This activity also compensated for the lack of LSA data. This detailed format for the CTASC-II task selection matrix was then used by the training developers to immediately begin development and production of such documents as the POI, lesson plans and presentation materials.

Worksheets. The Learning Analysis Worksheet (LAW) and the Job and Task Analysis Worksheet are normally developed once the task selection matrix has been finalized and approved. The LAW is a tool that is used for identifying the supporting skills and knowledges of task performance that must be acquired before the trainee can demonstrate mastery of a task. It includes performance steps, the time to perform each performance step and the method or methods that are used to teach each performance step. The Job and Task Analysis Worksheet is another tool that is used to record pertinent data about a task to be trained. This data includes conditions under which the task will be performed, and the standards that determine successful completion of the task. The Job and Task Analysis Worksheet also lists specific cues that will cause the task to be performed. These cues include items such as equipment being non-operational or equipment being damaged.

These documents are normally developed for each task on the task selection matrix. A typical development process for the LAWs and Job and Task Analysis Worksheets could take as long as 6 to 12 months for a standard acquisition item. Because the task selection matrix had the necessary data which could be used to produce lesson plans and presentation media and because the tasks had been validated on the actual NDI equipment, the CTASC-II LAWs and Job and Task Analysis Worksheets were not initially required. This resulted in a six-month development time savings. All worksheets were subsequently developed to validate the lesson plans. These documents were completed and delivered to PM TACMIS and the Computer Science School in December 1989.

Training Aids and Devices. To determine the best utilization for CTASC-II classroom training aids and laboratory training devices, the CTASC-II operational scenario and the trainee requirements were analyzed. During CTASC-II daily operations, one or two operators perform at a time. However, during a training class, there could be a maximum of twelve students; therefore, equipment was needed in the classroom to insure that each student could view the task explained and performed by the instructor. To fulfill the classroom training aid requirement, two wooden training devices were utilized. The wooden devices depict, but do not include, actual components of the Operations and CPU shelters. In addition, each training device portrays the signal entrance boxes and power entrance boxes for each shelter. These training aids took only one week to develop and cost approximately $2,000.

It was determined that a combination of simulators and the actual system components would best meet the laboratory/device training requirements to deliver both operational and maintenance training. Because of the low cost of the NDI equipment, it was economically feasible to utilize the actual system for most of the operational "hands-on" training. In addition, the fault isolation of LRU failures using software diagnostics was also taught on the operational system. Therefore, CTASC-II crew operational training and fault isolation were performed on the actual system.

Maintenance removal/replacement training was accomplished using a training device which teaches the crew members how to remove/replace failed components yet minimizes the damage to operational equipment. Six equipment racks were built similar to the equipment racks from the operational system. Because removing and replacing operational components in a training environment will often times cause true failures, failed components from CTASC-II systems were mounted in these racks for removal/replacement training. This device insured that damage to operational equipment would be minimized while providing full remove/replace training. The maintenance training device and additional equipment racks were designed and produced in four weeks at a cost of approximately $20,000 compared to the actual system cost of approximately $515,000.

In both training devices, redundancy of NDI equipment is not as significant to operational fidelity because the switch from the primary equipment to the secondary equipment is automatic and requires no operator actions. Therefore, the requirement to match the redundancy of the operational system is a design decision that each program office makes. It certainly is true that the fidelity of the operational device will enhance trainee job performance; however, not all redundancy must be functional during training. In the case of the redundant disk drives that are used in the CTASC-II system, once the trainees have learned how to partition one, two, or three disk drives and how to specify device files, very little is gained in partitioning all fourteen. Training focuses on how to shift from one drive to another when a failure occurs. Likewise, in maintenance training, once a trainee has removed and replaced one or two disk drives, it is not necessary to remove all fourteen.

Development

Lesson plan development for the CTASC-II crew training course began immediately after the task selection matrix was finalized. The lesson plans provide the instructor with an exact guide of how to present each block of instruction instead of the typical lesson plan outline, which provides topics and aids but requires the instructor to research the detailed subject matter for presentation. The CTASC-II lesson plans have detailed instructions, examples, and notes about the content and presentation of the class. Typical notes would include cues when to show a vugraph, when to ask students a question, or when the instructor should perform specific software commands.

This development process took ten EER employees three plus months to complete. The time required to generate the lesson plans was significantly reduced by the availability of the NDI documentation and the availability of a CTASC-II system to assure that lesson plans were presented properly. In addition to the lesson plans, all of the classroom presentation media were also developed in this three-month period.

Course Content. The CTASC-II crew training course consists of sixteen annexes with 424 classroom hours of instruction. The subject matter and associated hours are shown in Table 1.

TABLE 1: CTASC-II CREW TRAINING COURSE

ANNEX	TITLE	PERIODS OF INSTRUCTION
A	Overview	8
B	Security	2
C	Safety	2
D	Emplace/Displace	20
E	UNIX OS	40
F	Activate/Deactivate	20
G	Replace Consumables	4
H	System Administration	64
I	IDIS Minis Menu System	44
J	Bourne Shell Programming	48
K	Communications	36
L	Diagnostics	40
M	Multimeter Operations	4
N	Troubleshooting	24
O	Maintenance	36
P	End of Course Comprehensive Exam	32

Implementation

The first CTASC-II training system class was conducted five months after the beginning of the SAT process to develop the CTASC-II crew member course. This was the first presentation of training in a Training and Doctrine (TRADOC) command school which had been developed for a crew operated system. All other courses in the TRADOC schools are designed for individual completion.

This first class began on September 18, 1989, and ended on December 3, 1989. Students in this class were PM TACMIS personnel, STAMIS developers, and Communication Electronics Command (CECOM) Logistics Assistance Office (LAO) personnel. Many of these personnel were already working in the CTASC-II program without having any training on the system. Members of this class were not the target audience of the MOS 74D; therefore, additional remedial training was required by many of the students to satisfactorily complete the training.

The first class presentation to an actual crew began on January 8, 1990. This class progressed quickly and scored very high on the final test because each student was a 74D or 74F with a proven background in ADPE and CE.

A total of four crews have been trained in the nine classes presented to date:

- two crews from Fort Bragg
- one crew from Fort Hood
- one crew from Korea

A fifth crew from Germany began training on October 8, 1990; however, they were recalled on November 11, 1990, to their unit to be deployed in support of Operation Desert Shield/Storm.

The CTASC-II crew training course is being delivered approximately five times a year. It varies according to PM TACMIS scheduling. When the CTASC-II goes into full production and fielding, the class will be presented eight times a year.

Evaluation

EER Systems was not specifically tasked to perform an evaluation of the CTASC-II crew members as they performed their jobs. Such an evaluation could be used to identify required changes to the training that would enhance performance in the field. However, the Persian Gulf War provided an opportunity to evaluate crew performance in an actual operational environment. When Operation Desert Shield began, the CTASC-II SARSS system was undergoing Software Acceptance Test (SAT) at Fort Bragg; the TAMMIS SAT at Fort Hood had not yet begun. Two crews had been trained to operate and maintain the systems, one designated for the SARSS system, the other for the TAMMIS system. When the decision was made to transport these systems and crews, as well as the contractor support, to the Kingdom of Saudi Arabia (KSA) to complete system testing, the opportunity arose to evaluate the crews' performance and relate that performance to the training.

The SARSS system was installed in KSA on October 1, 1990, and the TAMMIS system was installed on October 15, 1990. The SARSS system was operational for 177 days and the TAMMIS system was operational for 166 days prior to returning to the United States. During that period, the crews were able to fully operate the systems on a 24 hour-a-day basis. Both the SARSS and TAMMIS users were provided the system administrator services required to process their information. When hardware failures did occur, the crews were able to utilize the redundant (backup) NDI equipment to continue uninterrupted service to the STAMIS users.

CTASC-II systems are considered to be mission available when the STAMIS users are able to process information with no disruption in service. Both systems surpassed the required mission availability requirement of 96%. In total, 46 maintenance actions were required during 9252 hours of operation with 24.7 hours of non-mission capable time which equates to a mission availability of 99.8%. This success was due in part to the crews' ability to quickly repair the systems in accordance with (IAW) the maintenance allocation chart (MAC) and when required, responsive contractor maintenance. Specific crew actions which helped achieve the required mission availability included:

- Regular Preventive Maintenance Checks and Services (PMCS) were performed which eliminated multiple ADPE problems and
- Removal/replacement of failed LRUs as required.

The performance of the crews during Operation Desert Shield/Storm demonstrated that the training was effective and that the crew members could successfully perform their jobs.

SUMMARY

The CTASC-II NDI Training System Acquisition was able to keep pace with the operational system acquisition because of several time saving factors:

- NDI hardware documentation was available to assist the task inventory.
- a detailed task selection matrix was developed by SMEs and contractor per-sonnel that significantly reduced lesson plan development time.
- analysis, design and development events were combined to expedite training system development.
- NDI equipment was available to assist task selection and laboratory training.

- NDI equipment cost was sufficiently low to utilize the operational equipment as a training device which permitted rapid procurement and installation.

ABOUT THE AUTHORS

George S. Peratino, Jr., is a Senior Systems Engineer with EER Systems Corporation. He manages the worldwide maintenance of the Army's Corps/Theater ADP Service Center-Phase II (CTASC-II). Mr. Peratino also provided logistic support during the early acquisition of the NDI CTASC-II systems. He has a Bachelor of Science degree in Electrical Engineering from Virginia Polytechnic Institute.

Robert M. Cowell is the CTASC-II Training System Manager for EER Systems Corporation in Augusta, Georgia. Mr. Cowell has over ten years experience in training system acquisition. He has a Bachelor of Business Administration degree in Management from Augusta College and is a candidate for a Master of Business Administration degree in Management. ■

TRAINING ANALYSIS - PANACEA OR PLACEBO ?
THE UK ROYAL AIR FORCE EXPERIENCE

Wing Commander R M Prothero MRAes MBIM RAF

ROYAL AIR FORCE

ABSTRACT

In the mid-80s, the need for structured analyses of training needs prior to military training system acquisition became generally accepted. In 1988, the UK Ministry of Defence endorsed a policy statement that made it mandatory for the RAF to conduct a Training Analysis (TA), or Training Needs Analysis (TNA), for all future equipment projects.

Based on the apparent success of TAs conducted for other Forces, the RAF expected a great deal from the early studies set in hand. These included studies of the training needs for the EFA, the Tornado (Night Attack Variant), the Sea King helicopter, the Hawk and Jetstream visual system requirements, and a CBT project. Two of these TAs were let out to two contractors at the same time so the results could be compared.

Having received the results of all but one of the studies, it is apparent that none actually provides the Staffs with what was expected - a clear statement of the training organisation and strategy needed (where appropriate), and the training devices required to meet the training needs identified.

The reasons for this vary, some of the root causes are complex, but neither industry nor the RAF were at fault - what was lacking was case experience. However, the studies were not totally wasted. This paper will examine each of the TAs to extract the lessons learnt. Based on the Author's experience, it is believed that TAs are better conducted by non-military agencies under most circumstances, and it is essential that contactors have clear and agreed Terms of Reference for the studies. It is also essential for the military to provide full time Subject Matter Experts (SME) for major studies, and for the military training management to maintain a continuous overview of the progress of the study in relation to project assumptions. In addition, regardless of formal contract timescales, it must be accepted that it is better to accept a contract overrun without penalty, or take an iterative approach to the study, and get it right, rather than get it wrong.

Finally, it is expected that new lessons will be learnt from each TA in the future. The RAF expects to commission 6 major weapon system TAs in the early 90s. The scale of investment in synthetic training devices makes it essential that the refining of the TA procedure is a continuing process, and that the necessary feedback mechanisms are established amongst project personnel to enable this to be done.

INTRODUCTION

The aim of this paper is to relate the Royal Air Force's experience with Training Analyses (TA), or Training Needs Analyses (TNA), and draw appropriate conclusions. It will be suggested that the military and industry still have much to learn about the conduct of TAs. Although this paper refers to UK project analyses, there is little doubt that some of the lessons will be applicable to NATO weapon system programmes now, and in the future. Indeed, this view is supported by the conclusions of the Paper on the conduct of the TA for the C 17 Contractor Training System presented to the IITSC in 1990, which indicated that the RAF shares with the USAF many problems in determining training needs, and the optimum organisation for future system programmes.

RAF TRAINING ANALYSIS REQUIREMENTS

In the mid-1980's the requirement for a structured analysis of training needs prior to any military training system acquisition became generally accepted in the RAF. In 1988, the UK Ministry of Defence (UK MOD) endorsed a policy paper which made it mandatory for the Royal Air Force to conduct a training analysis as part of all future major equipment projects. Analyses were to be conducted early enough to ensure that all the necessary training equipment was delivered concurrently with the major equipment, or even before the aircraft system. The studies were to consider both aircrew and groundcrew input standards, and the optimum training syllabi and course lengths necessary to meet specified output proficiency standards. From this work, and data on future system deployment plans, and the number of squadrons and conversion units, it was anticipated that, after co-ordination with ground training requirements, the optimum training organisation could be specified. From this, the supporting synthetic training equipment could be defined and quantified. In relation to training equipment, it was the goal of the UK MOD to conduct the maximum amount of training on the least costly devices, and thus obtain the most cost effective training systems possible.

KEY STUDIES

The RAF expected a great deal from the early work based on the apparent success of

TAs conducted for other Forces. Studies were undertaken to determine the training needs for the:

1. European Fighter Aircraft (EFA)

2. Tornado Night Attack variant

3. Sea King helicopter

4. Hawk/Jetstream training

5. CBT for Tornado training.

From this list, it can be seen that the RAF has gone beyond just applying the analysis procedure to major projects alone. Two of these training analyses were let out to two external contractors concurrently so that the results could be compared, and confidence gained in the methodology. Whilst the EFA study was established in a considered manner, some studies were undertaken at very short notice to meet budget timescales. In addition, the sponsors for the studies varied.

To date, the results of all but one of the studies have been received, and it is apparent that whilst they contain much valuable data, none has actually provided the Staffs with what it ultimately needed - a clear statement of the training organisation and strategy (where appropriate), and the optimum mix of training devices required to meet the identified training needs. This lack of success should not be attributed to inadequate terms of reference or contract specifications for the studies, nor the organisations that conducted the studies. Rather, the cause was the lack of experience in conducting such studies amongst the various sponsors and UK contractors. Users and Analysts alike were at the bottom of the learning curve for such work; therefore, it was hardly surprising that some aspects of the TAs left something to be desired. What experience existed was spread thinly amongst sponsors, which points to it being desirable to have a single TA sponsor for the majority of aircrew studies. Furthermore, initially the RAF User was not able to capitalise easily on the analysis experience accumulated by the United States Military Forces and Industry on classified projects, nor indeed initial work in other NATO forces. In this latter area, it became apparent that the methodology and policy for TAs varied widely; not a satisfactory situation in multi-national programmes. Additional problems arose as a result of the fragmentation of a major corporation in the synthetic training industry, denying the UK access to corporate TA experience. However, later initiatives are helping to resolve this lack of communication.

EUROPEAN FIGHTER AIRCRAFT (EFA)

The EFA training requirements were examined by two external contractors. The terms of reference assumptions were, of necessity, lacking detail as EFA was a quadrinational programme in its infancy - not a good point to start on the first UK major TA. Many important decisions had not been made on the exact aircraft fit, role, deployment and indeed, the basic training strategy when the studies were started. After considerable preliminary work by the MOD scientific staff, the detailed study of UK training requirements was produced over a period of 9 months. The timescales were driven by the need to coordinate the TA with the views of our three NATO partners. This in turn had to be done in time to let the synthetic training equipment contracts to the aircraft prime contractor, to achieve concurrency of aids and aircraft.

As assumptions for the studies were translated into decisions, an iterative approach to upgrading the studies was taken. The first lesson here is that if more than one contractor is involved, some degree of standardisation of the format of the study is essential, or else the refinement and correlation of the TA results becomes excessively time-consuming. Moreover, it became apparent that without some degree of standardisation between the participating NATO countries on analysis formats and methodology, coordination was doubly difficult. Subsequently, the results of the study indicated that four major types of synthetic training equipment would be required: simulators (Basic, Flight and Tactics, and Full Mission simulators), trainers, classroom aids, and specialist trainers. This represents a major investment to achieve a cost effective training system, but already some study results have been called into question.

It has been suggested that it might be cheaper in the longer term to delete the Basic Flight Simulator and buy more of an advanced device, such as the Flight and Tactics Trainer, and so achieve economy of scale in their acquisition. Such a move would also build more flexibility into the training organisation. Flexibility is a prime criterion in a training system if it is to be effective in use, and must be quantified in some way in the TA terms of reference. Training cannot stop while hardware or software is modified or upgraded; thus, there must be some overlap between devices Moreover, once presented, it is counter productive to make pragmatic changes to the TA recommendations unless it can be shown that the work did not address fundamental cost issues - such as production run costs. If major changes are made to the requirement for training aids outside the TA, there is every chance that the cost effectiveness of the system will be degraded. It is also of note that the EFA TAs were conducted without any direct input from the Staff responsible for statements of ground maintenance personnel training requirements. There were reasons for not including the ground training requirement in the initial study. However, coordinating air and ground training requirements is going to be a long and difficult job. There can be no doubt that the ground training requirements should have been formally included in the study from the start. The relevant

policy staffs might not have been able to contribute much initially, but the lines of communication would have been established to ease the later linking of requirements. This process is now taking place, albeit at a slow pace.

TORNADO

The Tornado aircraft is due for a major sensor upgrade in the mid-90's, and so it was decided that a TA was required to determine the optimum training system beyond 1995, and thus the synthetic training device requirements. Whilst the analyses, again conducted by two separate contractors, would undoubtedly have produced cost-effective training systems, neither study is now useable because the assumptions in the contract specifications were outdated both during and after the studies. In addition, a decision made by the policy staffs on the positioning of night vision goggle (NVG) training within the training patterns of our crews, altered the required output standard of the training system. The lack of full-time Subject Matter Expert (SME) to carry assumption changes to the contractors was a fundamental problem, compounded by tight contract requirements.

Turning back to the policy staffs decision on NVGs, if an SME had been appointed, he would have been in an ideal position to provide educated advice to the staffs, thus preventing arbitrary decisions being made over NVG training without a full knowledge of the technology being made available to solve the perceived problem, and the long term consequences of any such pragmatic decision. In sum, contractors need some flexibility within analysis contracts to take into account change. In addition, as the C17 TA work supports, the provision of full-time SMEs is essential to the proper conduct of TA for major equipment programmes to allow the eventual User to monitor the progress of the studies, and ensure correct assumptions are used.

Nevertheless, there is a possible danger in excessive SME participation in TA studies. The results of a TA could be excessively influenced by the past experiences of the SME to the detriment of innovative training solutions being proposed which reflect modern teaching methods and technology.

HAWK AND JETSTREAM

In 1988, planned upgrades to RAF Hawk and Jetstream training systems, albeit subsequently delayed by budget considerations, resulted in the commissioning of TAs whose primary objectives were to determine the optimum visual systems to be employed on the simulators, and the impact of any synthetic training equipment upgrades on our training syllabi. In the event, over-reliance by study teams on advice from those already instructing on old equipment resulted in very narrow and unimaginative recommendations being made. Moreover, the proposals would have been excessively costly in one case, and inappropriate for student long term training objectives in the other. These problems could be put down to the inexperience of the contractor staff in dealing with such a military training study.

OTHER STUDIES

It is not the author's intention to make it sound as if the performance of contractors in the conduct of TAs has been unsatisfactory, nor that the varied sponsors were less than diligent. Rather, experience indicates that external contractors must be employed for TAs. Whilst the conduct of studies must not rely too heavily on training psychologists, financial managers, or academics, and the major input must always come from those who have practical experience of aviation training, only contractors can devote themselves full-time to an analysis task, develop broad based TA methodology expertise, and provide the necessary psychologist specialists and widely experienced training analysts that any TAs require. They can also provide stable teams for lengthy or iterative studies. Military personnel move on re-assignment or promotion, and so maintaining analysis skills, or consistency of approach, would be very difficult if we relied on RAF or MOD staff internal studies alone. Also, only contractors are likely to hold the necessary current technical expertise and resources to conduct TAs in the future. As pressure grows on the military to reduce manpower levels, this will be especially true. Moreover, the peaks and troughs in overall TA work requirements indicates that it would not be cost-effective to rely solely on established analysts in military training organisations. This view on the use of contractors is supported by experience with an urgent short term study into RAF Sea King training needs, and a TA into the future CBT needs of the Tornado GR1 aircrew ground school training. The former study was completed, but needed considerable revision before a project milestone, but the staff resources were no longer available as new and higher priority projects had emerged. The Tornado study work ceased immediately when the aircrew staff officers were re-assigned to Gulf War duties. It is fortunate that a delay in this project will not affect long term plans significantly. Nevertheless, the foregoing does not rule out conducting TAs using military resources under certain circumstances. For example, such a course of action might be desirable if the analysis work would not be great; it is a revision of an existing training system, or time is of the essence.

Following this, after saying that the RAF experience shows that contractors should normally be responsible for analyses, another difficulty has emerged. The terms of reference, or contract specifications

for TAs, must be very carefully written, User orientated, and fully understood by both the analysis contractor and the User. It is not necessary to describe fully another visual training requirements analysis conducted for the RAF, but suffice it to say that, despite the best efforts of the contractor, who met the contract requirements, the output was in a totally User-unfriendly form. The conclusions did not help the relevant staffs to develop a system requirement, and when the shortcomings of the study format were appreciated, it emerged that there was no flexibility in the contract timescale to allow the format of the study to be modified. Finally, during this study it became apparent that as the contractor wished to retain the copyright of the final report, which was considered proprietary, even if a visual system specification had been included, it could not have been used directly in the acquisition process. This will be quite unacceptable for the UK MOD in future as the final report of a TA will be fundamental to the competitive acquisition of synthetic training aids.

CONCLUSIONS AND RECOMMENDATIONS

The following are the conclusions based on the RAF's experience of TAs to date:

1. TAs are better sponsored by a single User office to ensure consistency of management and procedures.

2. There is a need to develop a standard format and methodology for training analyses, not only to assist potential contractors, but also to allow NATO partners in joint programmes to cooperate fully.

3. Changes to any TA recommendations must be fully cost justified, and the impact of change on the cost effectiveness of the total system assessed.

4. The requirement for training system flexibility to be recognised in analysis recommendations, and must be quantified in some way in TA specifications and reports.

5. Full time subject matter experts are essential to any major training analysis, but they must not be part of the formal analysis team.

6. TA contracts must allow the study contractor some leeway to absorb changing circumstances during the period of the study. An iterative approach to the development of TAs should be recognised as an acceptable option.

7. Industry should be contracted to conduct TAs unless the studies are short, time is of the essence, or existing systems are being upgraded.

This list is not exhaustive, and the UK MOD will no doubt be learning new lessons from each major analysis that is conducted in the future. However, quite apart from the lessons learnt during recent TAs, there will probably be lessons to be learnt from the acquisition and implementation phases of past TAs.

As yet, no project that has been through the full TA process has reached the field, nor is there any formal process to assess the success or otherwise of a TA. However, in the longer term, a feedback mechanism must be established. Indeed, a start has been made to establish such a mechanism through the work of our national research agencies.

Also lacking, at present, is some formal network for Users to communicate on TA matters within NATO. However, this situation is also changing as European cooperation into aircrew training research is appearing under EUCLID programmes. With so many diverse projects ahead, a great deal of money could be saved if the training problems were tackled together. In this area, more NATO cooperation when studying the training needs of forces in multi-national programmes could be very cost-effective. Much could be learnt and shared from such projects as the C17 contactor training system, the C130J if it becomes a reality, the US Navy's P3 replacement programme as it develops, and upgrade programmes following the cancellation of new advanced weapon systems in the USA. The RAF has developed a project orientated TA management procedure and TA format which could be of interest to other forces. In addition, the Service is now becoming more interested in contractor training systems that would require TAs to be conducted as part of the implementation process.

Finally, potential TA contractors should note that the Royal Air Force expects to commission six major weapon system analyses in the early 90's. The scale of investment in synthetic training devices makes it essential that the refining of the TA procedure is a continuing process, and as stated earlier, that the necessary feedback mechanisms are established amongst project personnel. Applying a formal methodology to training system needs is essential to promoting cost-effective training, but the results must not be implemented without a thorough User assessment of its recommendations to ensure it will be flexible in operation, and capable of fine tuning as working assumptions, or role employment, change as the project matures. To go back to the introduction, and the title "Training Analyses - Panacea or Placebo", the major point that has emerged from RAF TAs so far is that such studies do not provide a foolproof answer to training problems, but they do go a long way towards giving us the ability to set up training organisations with a fair degree of confidence in their performance, and effectiveness. They are a useful tool - but their utility depends on the hand that grasps the handle. The User must also be training system "intelligent". By sharing our experience, we should be able to guide our future project TAs more

accurately based on previous experiences, knowledge and common sense.

REFERENCES

"Pilot Training for EFA - Getting it right". Parfitt IITSC 1990

EFA Training Analysis Reports
UK Crown Copyright

Tornado GR4 Training Analysis Reports.
UK Crown Copyright

Hawk/Jetstream Visual Requirements
Training Analysis Reports
UK Crown Copyright

UK MOD Air Force Board Standing Committee Paper AFBSC(88)15(X)

The C17 ATS. Various IITSC 1990

ACKNOWLEDGEMENTS

The author wishes to thank Alan Parfitt of the UK Ministry of Defence, and Chuck Sturges of Ball Systems Engineering Division for their help and encouragement.

THE AUTHOR

Wing Commander Bob Prothero holds the post of TO3(RAF) in the UK Ministry of Defence where he is responsible for the management of RAF flight simulators, and providing advice to the Policy Staffs on the application of synthetic training technology to future weapon system projects.

He is a graduate of the Royal Air Force College; is rated as an exceptional pilot and instructor, and has held a wide range of flying and staff appointments. He is a Member of the Royal Aeronautical Society, and British Institute of Management.

DESERT STAARS: SUSTAINMENT TRAINING FOR ARMY AVIATION READINESS THROUGH SIMULATION

Parker R. Goodwin and Samuel Knight
CAE-Link Corporation
Binghamton, New York

CW4 Robert Monette, U.S. Army
Fort Rucker, Alabama

ABSTRACT

As last year's I/ITSC papers were being written, the prospects for world peace were having significant effects on military strategies and priorities. Many analysts, however, were cautious. Their concerns were based primarily on the potential for low-intensity and regional conflicts. A paper proposing the need for advanced mission training and rehearsal (Monette, et al.[1]) noted that "while uncertainty surrounds the perception of a diminished threat of world war, there is l.''ɔ ʇuestion that there exists an inevitable threat of armed conflicts with radicals, revolutionaries, terrorists, and drug cartels. It can also be anticipated that the severity of these conflicts will continue to increase..." As described in that paper, the non-conventional nature of such conflicts has resulted in increased war fighting emphasis on timeliness and precision. To support training in these skills, new concepts were proposed, including a recommendation for integrated mission training and rehearsal facilities. These facilities would employ advanced simulation technologies and specialized training programs which would be dedicated to enhancing the mission readiness of aviation crews. By the time I/ITSC '90 commenced, events in the Middle East had significantly reinforced the need to pursue such advanced training capabilities.

The previously referenced paper also noted that it would take teamwork to meet the changing military training environment for the 1990's-teamwork between users, military planners, analysts, and industry. This year's paper is intended to discuss such a team and the program implemented by that team to develop the sustainment and mission-similar training capabilities proposed in the 1990 paper.

INTRODUCTION

Military aviators receive primary skills training during their first few years of service. The main training emphasis during the rest of their career of possibly sixteen or more years is to enhance and sustain what was initially learned (Miller[2]). While ground-based simulations have been used extensively and effectively for primary training, the military has historically resisted utilizing the same types of assets for sustainment training. A reason often noted is the ever present concern that actual flight time will be reduced.

Recently the United States Army deployed to the Middle East to assist in executing a United States policy and a United Nations directive. As the timeline for deployed aviation units stretched into weeks and months, the unit commanders were forced to enhance and sustain the combat skills of their flight crews with the only asset available that could meet the training challenge. combat aircraft. This situation brought to the forefront an age-old problem of military command: How much of my combat assets do I allocate to training, and what additional burdens will my logistics system bear in terms of ammunition, fuel, parts, etc.?

The challenges of Desert Shield dramatically showed a specific need for deployable sustainment training devices. However, it is becoming widely recognized that the basic requirement for such devices exists even during peacetime.

MISSION-SIMILAR TRAINING

In our 1990 I/ITSC paper (Monette[1]) an integrated mission training and rehearsal program was proposed which would include dedicated resources to support an extensive hierarchy of advanced crew readiness training. This hierarchy, as shown in Figure 1, is divided into mission graduate sessions and mission-specific sessions. Figure 2 illustrates that the mission graduate sessions are intended to produce and maintain mission-capable crews, while the mission-specific sessions serve to optimize crew readiness for specific missions.

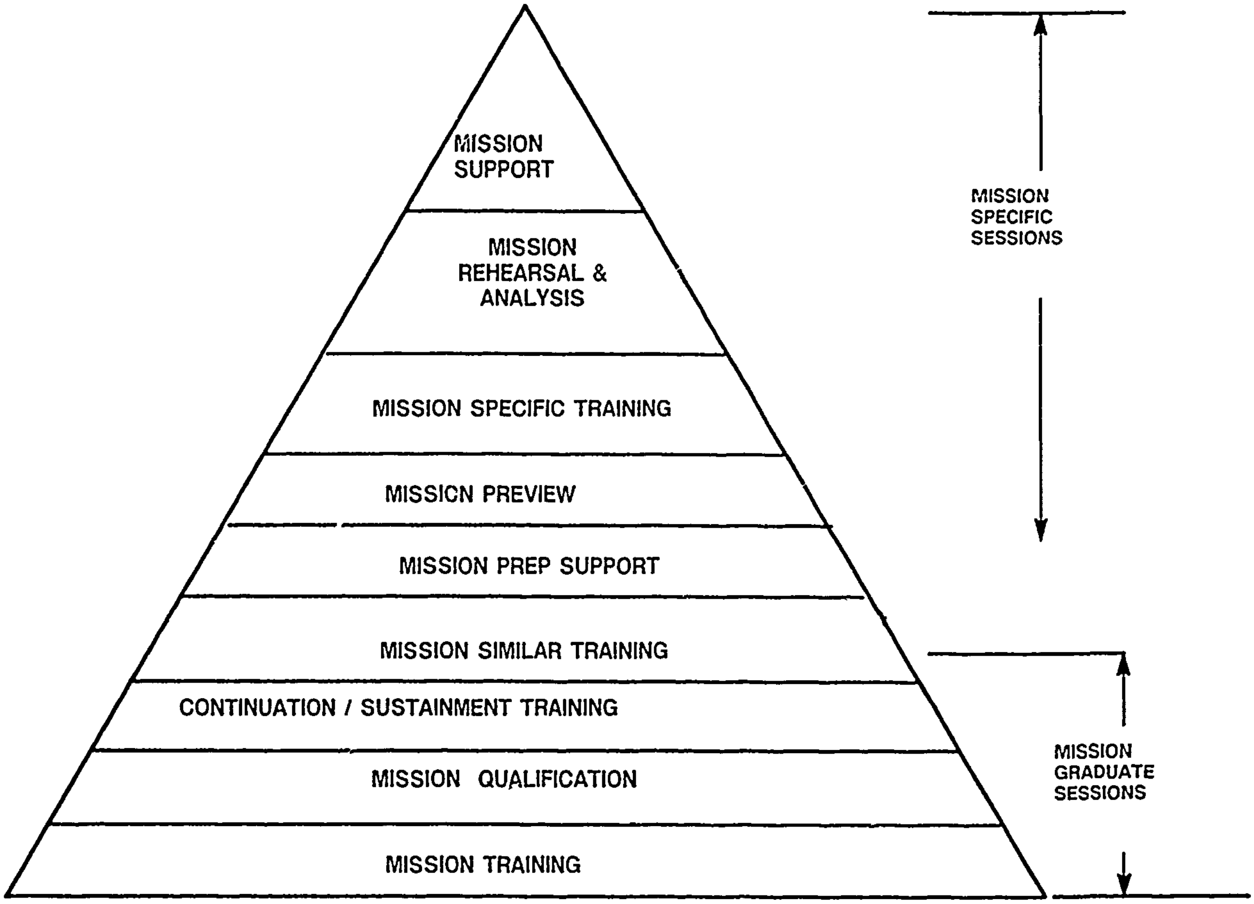


Figure 1 Mission Training/Rehearsal Hierarchy

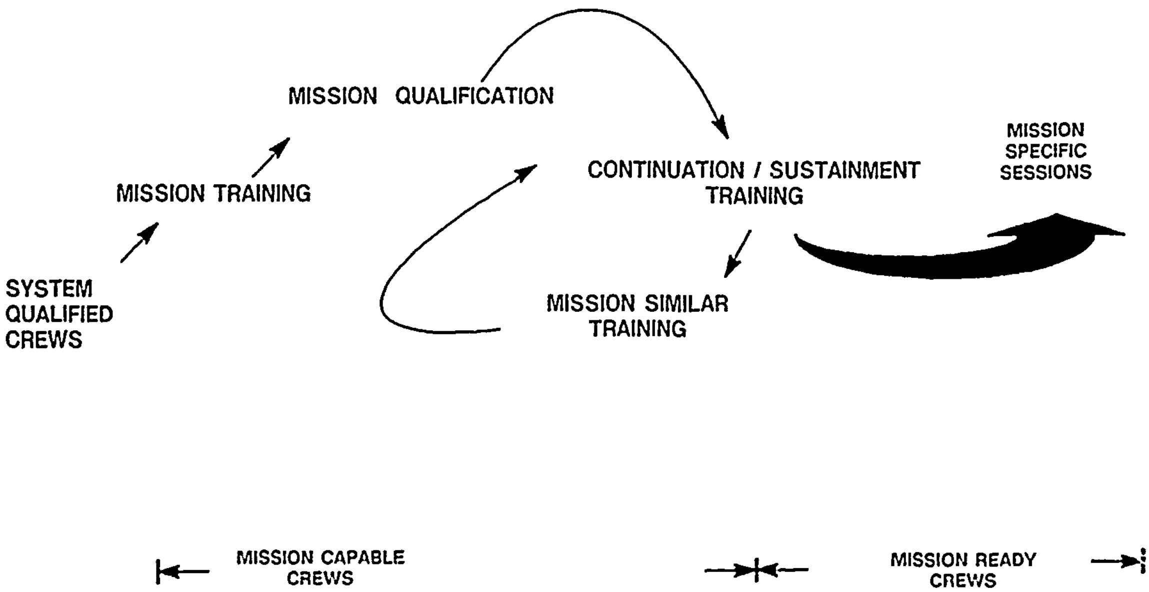


Figure 2 Mission Qualification

Mission-similar training is a significant element of the proposed hierarchy. As shown in Figure 1, it is applicable to both mission-specific and mission graduate sessions. When specific operations are required, crews would practice anticipated types of missions using databases of (or similar to) the designated mission area and employing tactics based on knowledge of the identified threat. This type of training would be employed until more detailed mission plans and associated mission rehearsal databases could be made available. Mission-similar training, however, is also intended for use as an extension of sustainment training. More specifically, crew preparedness can be enhanced by mission-similar training relative to areas of "potential" activity.

As global situations develop, it is very possible to anticipate the need to train for conflict in an unfamiliar type of terrain against an unfamiliar threat. The technology exists to rapidly construct geospecific visual and sensor databases and to populate those databases with appropriate threat arrays. These databases and threat arrays, coupled with the additional training capabilities inherent in advanced conflict simulations, provide an optimum capability for mission-similar training. Utilizing these technologies provides a means by which potential crisis situations may be prepared for in a general sense. It should be further noted that when the capabilities of these technologies are applied to an exact area and threat, the generalities of mission-similar training become the specifics of mission rehearsal. The unique blend of technology and task complexity required for mission training and rehearsal has been successfully met by a program called Desert STAARS (Sustainment Training for Army Aviation Readiness through Simulation).

DESERT STAARS OVERVIEW

The need to be prepared to fight and fly in different types of terrain against varying threat arrays was pointed out in the recent Middle East war. Most of our crews were trained in the techniques and tactics appropriate for wooded temperate zones populated with a specific type of threat. The requirements imposed on them to modify their existing tactics and to develop new desert tactics on the fly demonstrated in the strongest possible terms the need to provide mission-similar training prior to their arrival in the area of conflict.

This need, coupled with the overall requirement for a sustainment training capability, provided the incentive for the U.S. Army's Desert STAARS program. Desert STAARS is a training modification to the Army's existing AH-64 Combat Mission Simulator (CMS). This program employs advanced mission training and rehearsal capabilities which were demonstrated (proof-of-concept) on August 4, 1990 (two days after the invasion of Kuwait). The primary new technologies involved include geospecific visual/sensor databases, Multi-Simulator Networking (MULTISIM) and Force Level Simulation (FLS). Desert STAARS applied these technologies to create the mission-similar training capabilities described in the previously referenced I/ITSC paper.[1] Desert STAARS development was completed in a very short 90-day period through the concentrated teamwork and dedication of PM TRADE, the Naval Training System Center (NTSC), the Directorate of Training and Doctrine (DOTD) at Ft. Rucker, U.S. Army Subject Matter Experts on aviation training, and CAE-Link and its supporting vendors.

SYSTEM REQUIREMENTS

Three primary systems areas must be addressed to facilitate mission-similar training capabilities. The first area is the simulation of the airborne weapons system in which the crews will train. The second area is the creation of an appropriate simulated mission environment. The third area involves providing system mission control and performance monitoring capabilities.

For Desert STAARS development the U.S. Army provided the AH-64 CMS, which is noted for its capability to accurately simulate the AH-64 and its mission performance capabilities. The Army also provided a UH-60 Flight Simulator (FS) to support testing of the system network modes.

The most extensive area of development for Desert STAARS was the mission environment. An obvious large-scale task was that of creating correlated geospecific visual/sensor and tactical databases. The government-specified 80 km by 100 km databases that were generated encompassed primary areas of tactical interest in Kuwait.

The geospecific visual/sensor database was created from numerous sources. The initial basis for this database was Defense Mapping Agency (DMA) Digital Terrain Elevation Data (DTED). This data was used to provide terrain boundaries and contour. DMA DTED was also used to provide a correlated FLS terrain database. The database was then supplemented with DMA Digital Feature Analysis Data (DFAD). This data provided information on such cultural features as international boundaries, city areas, airfields, major roads, power lines, oil fields, pipelines, water towers, and major buildings. All source data was supplied at Level 1. In those instances where DMA data resolutions were too

coarse, maps and photos were used to provide more detailed placement of features.

Both military and non-military maps were employed, with special emphasis applied to areas of tactical or navigational interest. Three examples of these areas of interest are the American Embassy in Kuwait City, a political prison, and the Kuwait City International Airport.

Numerous techniques were used to develop cultural areas. These techniques included the use of previously existing models, the implementation of 2D models, as in representing sewage treatment facilities, and the use of new 3D models. For example, in developing the Kuwait City International Airport, the only features provided by DMA were the two main runways and the main highways providing access to the airport. Photos were used to determine additional cultural requirements. Hangers and the main tower were implemented using previously existing models. New modeling provided additional access roads, taxiways, and the main terminal. Throughout the database phototexturing was used extensively to enhance the realism of cultural and terrain features. To allow extended ingress and egress operations, the geospecific database was supplemented to 300 km by 300 km with generic roll-on terrain on three sides and roll-on water on the fourth side.

To support tactical operations, ten new target types were added, at government request, to the AH-64 CMS existing inventory of 58 targets. Moving target pathways were added to the database and a new system capability was developed to allow relocatable fixed target sites. This was an important step in the evolution towards mission rehearsal capabilities since it allows sites to be rapidly changed in response to updated intelligence data.

To support unique desert flight requirements, special visual/sensor effects were added to simulate sand storms and blowing sand caused by rotor wash.

An important detail in allowing simulation of real-world operations is the provision for accurate, correlated navigational capabilities. (The correlation affects the position of the ownship and other vehicles as well as cultural features relative to real-world map positions.) Numerous navigational models of varying accuracy are used in different simulation systems. Engineering analysis, however, showed that positional errors in the thousands of meters could occur if a common system was not implemented. The World Geodetic System 1984 (WGS 84) was employed in Desert STAARS because of its accepted accuracy, its usage in military systems such as AH-64, the availability of accurate conversions to and from military grid systems, and the acceptance of WGS 84 for the forthcoming Distributed Interactive Simulation standards.

The previously referenced I/ITSC paper[1] noted that for advanced mission training and rehearsal to be effective, the environment of the mission must be closely replicated. This replication must transcend the traditional notion that allowing a flight crew to experience a set piece situation is mission rehearsal. True mission training and rehearsal requires two dissimilar elements within its environment to function properly. precision and chance. More succinctly put, to be effective the simulation must be precise in presenting the known details of terrain, weapons, enemy disposition, etc., and simultaneously allow the vagaries of combat. Hopefully this blending of opposites will allow the crews to safely experience the happenstance that is a part of tactical operations.

The key to replicating the "fog" of war in Desert STAARS is the FLS conflict simulation. In general the FLS provides a "thinking" type threat rather than an "if met" automated threat. This thinking opponent introduces the element of chance and sharply increases the realism of the training. The FLS creates a knowledgeable opponent by modeling not only the threat's parametrics but also its tactics, command and control structures, communications links, short- and long-term memory functions, and even misperceptions. A special feature added for Desert STAARS was a manual control allowing operator-controlled FLS kills. This provided an ability to simulate team fire support during missions where a networked wing man was not available. Another feature added was the ability to store and review FLS scenarios for crew debriefing.

Desert STAARS modifications were [illegible] implemented to enhance the previously ex[illegible] AH-64 CMS threat algorithm to be operable with the geospecific database and new threats. This feature allows instructors to conduct, when necessary, sustainment training in individual skills which may not be as easily concentrated on in the FLS total mission environment. This stand-alone capability enhances the flexibility of the training system and makes it more responsive to user needs.

Another important element of Desert STAARS is the MULTISIM network interface which couples the AH-64 CMS to the FLS. The network is designed to be expandable to allow other compatible training devices to interoperate with the AH-64 CMS and the FLS (e.g., devices such as the UH-60 FS). A

significant feature of MULTISIM is the ability to prioritize and sort network data going to the simulator. More specifically, in Desert STAARS the FLS scenarios can invoive up to thirty players, while the AH-64 CMS can visually display only ten of the players. An elaborate prioritization algorithm determines which targets are to be displayed during FLS operational modes. The algorithm considers such factors as threat lethality, target line-of-sight, out-the-window and sensor fields of view, weapon employment, and target range. A separate sorting algorithm is used for sorting threats to be processed by the EW (electronic warfare) simulations. These algorithms significantly increase the target density of the AH-64 CMS virtual battlefield.

Another area of development in Desert STAARS was that of creating mission control and performance monitoring capabilities. Modifications were made to the AH-64 CMS instructional pages to allow operation in the geospecific environment. These included page changes to allow mode and FLS scenario control as well as extensive digitizing to create the various cross-country and tactical map pages. Modifications were also made to implement the instructor capability to relocate fixed target sites, including provisions to observe the desired relocated position from various points of view before storing the new position.

A Tactical Operation Center (TOC) was also developed to allow observers to monitor Desert STAARS training scenarios. The TOC includes a large-screen monitor which graphically displays an overview of the FLS scenarios, including player interactions. The TOC also includes repeater monitors of the AH-64 CMS out-the-window and sensor imagery as well as a monitor to repeat selected map displays from the AH-64 CMS instructor station. Audio provisioning allows TOC communications with the AH-64 CMS and networked devices via a simulated radio link. The FLS digital voice for selected players can also be monitored in the TOC. The TOC is intended to provide a facility for commanders, other aviation crews, and mission analysts to observe and critique mission performances.

USER/ANALYST/ENGINEER TEAMWORK

To provide the capability for mission-similar training, especially under time constraints similar to Operation Desert Storm, requires the user, analyst, and engineer to work as a team. Each of the team members brings a synergistic skill to bear on the training problem. The user has the best intelligence-gathering capability. He knows which information is most important to his training, where it may be obtained, and has better access to the information. He knows what types of targets need to be on what sort of visual database to provide the mission-similar or mission-specific training he needs. The engineer, in turn, knows what type of information is required to enable the technology to support the training.

Streamlining the development of the system requires an on-site user Subject Matter Expert (SME). The SME, provided that he has decision authority, can radically shorten the time required to develop a training tool by expediting or eliminating Preliminary Design Reviews (PDR), Critical Design Reviews (CDR), and Progress Reviews. In the Desert STAARS program the rapid timeline was made possible by the use of two full-time SMEs. One SME was dedicated to the visual database and the other was dedicated to the tactical enhancements, including FLS.

Unit training programs are a blend of mission requirements and available technology. The user knows what skills are required to accomplish his mission and the training analyst knows how to employ the technology to best support the training of those skills. The analyst will also bring additional knowledge of the technology that could result in training enhancements in unexpected areas. Desert STAARS technology, incorporating a geospecific database, FLS, and relocatable target sites, provides Army aviation units the necessary tools to construct complex mission scenarios. These mission-tailored scenarios, under strict user control, will be instrumental in the accomplishment of the unit's sophisticated tactical training.

SCENARIO GENERATION

The training scenarios associated with mission-similar or mission-specific training fall within the design of the FLS. Scenario generation involves the definition of all participating players, player locations, command structure, communication nets, and areas of responsibility. These definitions, developed using any word processor text file, are interpreted by the FLS. Once FLS has digested the file, it is able to present operational information graphically to aid in scenario planning. As an example, the threat Air Defense Artillery (ADA) might be instructed to report all contacts to higher command and request permission to engage targets within its range. In such a situation, if the ownship enters into the acquisition zone of the ADA his position will be reported, but the ADA weapon system would not engage without permission from his chain of command. Five scenarios were generated as part of Desert STAARS development. The initial three were developed by the engineering team with SME support. The final two were developed by the SMEs.

This collateral learning is another example of the advantages of employing dedicated SMEs.

READINESS ASSESSMENT

Readiness assessment is the logical step to follow up any training advancement. The Desert STAARS assessment was based on mission-oriented design concepts (Stark[3]). Specific testing was conducted to verify the new features. Training readiness, however, was determined in the crucible of operational mission training using tactical scenarios.

LESSONS LEARNED

Precision, realism, and time are critical factors in preparing and presenting a mission rehearsal scenario. Perhaps the most difficult obstacle to overcome is time. Within this constraint you must achieve all other elements of the mission rehearsal requirements. To make the process of achieving such time-critical programs easier in the future, a few lessons are noted which were learned on Desert STAARS.

After the need and requirements have been determined through the Statement of Work (SOW), the teamwork begins. The first task is the data collection process. To support this awesome task, a centralized collection point should be established. Authorized agencies could use the one point of contact to "shop" for the individual data that would satisfy the SOW. The agency would be in constant touch with other subordinate agencies so as to facilitate continued buildup of more detailed databases.

Higher levels of DMA data should be made available. As stated, the Desert STAARS project was generated from Level 1 information. With the application of higher-detailed data a project would benefit in two critical areas: precision and time. A more exact geospecific database would be generated which would allow the crews to encounter the area in an even more realistic manner. Also, because more details would be provided from the DMA sources, the time required for tedious hand modeling would be reduced.

Another recommendation to allow generation of detailed databases in minimum time is to generate comprehensive library files. These files would contain enhancements that would be available for use throughout the simulation community. Typical files would include but would not be limited to natural entities (trees, shrubs, etc.), cultural objects (buildings, bridges, power lines, etc.), and environmental effects (blowing sand and dust, fog, rain, clouds, etc.). Advanced texture patterns would serve to add realism to the library files. These texture files should be continuously expanded to allow for added dimensions of realism.

An important task of Training Systems Engineering (TSE) is to display the mapping/training information to the Instructor Operator (IO) in the same exacting detail as the geospecific database is displayed to the crew. Accordingly, TSE should be closely involved with the database generation. Because the Desert STAARS program development was not completely collocated, the database and instructional maps were generated separately. In future programs techniques should be implemented to automatically extract map data during the database development process.

CONCLUSIONS

Our future opponents may not allow us the time to conduct mission-similar training, much less mission-specific training, once we arrive within the theater of operations. The tools that provide us that training capability need to be with our soldiers now. We must present our troops the opportunity to fine-tune their perishable mission skills and to continually stimulate the growth of those skills. The geospecific databases and tactical enhancements of Desert STAARS, when resident within available simulators, will provide our combat aviation crews the opportunity to sustain the specific mission skills required to successfully prosecute combat operations. In the pursuit of that skill sustainment it is also possible to discover mistakes or potential mistakes that are mission critical. Given a specific mission, this ability to analyze performance and detect mistakes prior to combat is, in actuality, MISSION REHEARSAL.

REFERENCES

1 Monette, CW4 Robert, Gary George, and Samuel Knight. "Mission Training And Rehearsal: Employing Simulation To Its Full Potential". 1990 Proceedings of the Interservice/Industry Training Systems Conference.

2 Miller, Dennis M.. "Foundations For Tactical Training. A Challenge To Industry". 1988 Proceedings of the American Institute of Aeronautics and Astronautics.

3 Stark, Dr. Edward, Gary George, and Samuel Knight. "Mission Oriented Simulator Development". 1988 Proceedings of the American Institute of Aeronautics and Astronautics

ABOUT THE AUTHORS

Samuel N. Knight is a Staff Scientist with CAE-Link Corporation in Binghamton, New York. Mr. Knight was the Program Engineer for Desert STAARS. He is currently the Principal Investigator for the Multiple Simulator Networking (MULTISIM) program, and was also a technical advisor on the LHX DEM/VAL program. He was previously the Tactical Systems Lead Engineer throughout the conceptual and development phases of the prototype AH-64 Combat Mission Simulator. He has over 20 years of experience in visual and tactical systems simulation. He has a Master of Electrical Engineering degree from Cornell University and a B.S. in Electrical Engineering from Clarkson University. Mr. Knight has published work in the areas of tactical simulation, simulator networking, simulator fidelity, and mission rehearsal.

Parker Goodwin is a Senior Training System Designer with CAE-Link Corporation of Binghamton, New York. Mr. Goodwin served in the U.S. Army from 1968 to 1988 and retired as a Chief Warrant Officer Four, Master Aviator. Since his arrival at CAE-Link in January of 1989, Mr. Goodwin has conducted analyses to determine the design requirements for training systems to support advanced technology aircraft, including the First Team's Light Helicopter (LH). He was also a training analyst for the Desert STAARS program. He holds an Associate of Science from the State University of New York and is continuing his higher education at SUNY Binghamton.

CW4 Robert Monette is a Senior AH-64 Instructor Pilot assigned to D Co. 1-14 Avn. Reg., Ft. Rucker, Alabama. With 20 years of experience and two aviation tours to Vietnam logging over 1400 hours of combat missions, he is a subject matter expert in attack helicopters and simulators. Mr. Monette was an Army aviation SME for the Desert STAARS program. He previously evaluated both the AH-1Q and modernized Cobra Flight and Weapons System Simulators from 1979-1982. Presently his special duties include being a member of the AH-64 CMS acceptance team and also an assignment as the AVSCOM New Equipment Training team chief for all CMS device training. He received the AAAA Trainer of the Year Award in 1988. He has an A.S. degree in Aviation Science from Embry Riddle University. CW4 Monette has published work in the areas of simulator networking, simulator fidelity, and mission rehearsal.

QUICK-RESPONSE TRAINING SYSTEM MODIFICATION AND ITS IMPACT ON ARMY AVIATION SUSTAINMENT TRAINING

CW3 Charles Fullmer
AH-64 Standardization Instructor Pilot
U.S. Army, Fort Rucker, AL

Ron Matusof and Victor Polkowski
CAE-Link Corporation
Binghamton, NY

ABSTRACT

"Train like we fight, fight like we train" is an age-old axiom of military training. It is a training concept which is easy to grasp and makes sense. However, this training concept presupposes that we do, in fact, know how we are going to fight. In the last two years, the world has undergone significant political, social, and economic upheaval. The military community has had to evaluate the identity and nature of the threat, develop an appropriate set of countermeasures to the threat, and then temper the plan with the fiscal realities of shrinking defense budgets. All of this has meant a change in many mission requirements which must be reflected in the training of the military. This paper discusses the issues of providing sustainment training for aircrews in the face of rapidly changing mission requirements. It discusses the role of political, economic, and technological impacts upon the defir.'tion of the threat and of the mission. It then discusses the differences between mission rehearsal and sustainment training and suggests the concept of quick-response modification of existing training systems for sustainment training. Finally, it discusses an actual implementation of the quick-response modification to support .apidly changing sustainment training requirements for Army Aviators.

INTRODUCTION

For the period of time known historically as the Cold War, western military and political doctrine was geared toward a defined threat (the Warsaw Pact) and a defined theater of combat (Europe). Consequently, during this period of time, weapon system development and the accompanying training was geared toward a potential military activity in the European scenario.

During this same period of time, the United States responded to at least 187 international incidents and crises, excluding the Korean, Vietnam, and 1991 Gulf wars.[1] More than ninety percent of these conflicts have occurred in Third World countries. During the Cold War, emphasis was clearly focused upon the Soviets. United States tactics were defined to combat a Soviet advance. The word "threat" became synonymous with the term "Soviet". This is not to say that the emphasis was in error or misplaced. General Carl E. Vuono has stated "...it is clear that the possibilities of direct U.S.-Soviet conflict are running at ebb tide and that our venerable strategy of containment has been victorious".[2] Today's Combined Arms Warfare (CAW) practiced by the tri-services is derived from our understanding of Soviet tactics and equipment capabilities.

Since late 1989, the political structure of the world has changed significantly. This change and the recent military actions in Panama and the Persian Gulf underscore that future CAW tactics must be modified to account for new threats and new environments, as well as for changes in a known threat's behavior and tactics.

The U.S. Army Staff Officer's Handbook states "the science of war is in a constant state of change, driven by new technological developments which can radically change the nature of the battlefield".[3] The science of war is also modified due to massive changes in the world political structure and economic pressures brought about by the fiscal realities of smaller military budgets. The introduction of new threats, often employing non-conventional warfare, also radically alters the nature of battle.

By 1995, the U.S. military structure will most likely be very different from the way it is today. Troop size is expected to be cut by upwards of twenty percent. Tactics will require modification to account for fewer troops, as well as changes in the threat's identity, technology, and doctrine, including countermeasures for non-conventional warfare. In response to the smaller size, each element of the CAW structure will have to take on additional or modified mission requirements.

Training requirements are derived from the mission requirements of units and crews. In a world with rapidly changing mission requirements, appropriate changes to training should also occur.

Unfortunately, the same political, economic, and technological pressures placed on military planners are also being felt by the training community. Additionally, the training community is feeling pressure from environmental issues which make the live practice of tasks such as low-level flight and weapon delivery impractical. Training must not only account for rapidly changing requirements, but must do so efficiently and with consideration to the availability of assets which may be affected by budgetary and environmental constraints.

The change in mission requirements will be felt most by mission-ready crews. During peacetime, crews participate in sustainment training to maintain proficiency in skills which they will need during wartime. Peacetime training, however, is derived from the perceived threat at the time the training requirements are developed. History has shown that actual combat tends to occur against a threat operating under conditions which were not fully accounted for during the sustainment training. The ability to quickly modify training to account for new threats or combat conditions is a major challenge facing the training community in the 1990's and into the 21st century.

SUSTAINMENT TRAINING

Most crews in the armed forces are fully qualified to operate within a wartime environment. These crews are kept at a high state of readiness through sustainment training. Sustainment training is intended to maintain a crew's skills, developed during previous training exercises, at the highest possible level. In a world without change, this can be done with relative ease. Without change, training requirements and systems can be established and refined over time to tightly mesh the crew's skills to the intended mission. Rapidly changing requirements radically alter the face of sustainment training.

The effort of defining combat-ready tasks (tasks in which proficiency is tantamount to combat readiness) and the method of training and maintaining proficiency in those tasks is relatively academic. The AH-64A Combat Mission Simulator (CMS), as a case in point, was designed with a European stylized terrain data base specifically to train those tasks deemed necessary by the training community for combat readiness. But the changing nature of the threat has added a requirement to sustainment training that it provide a means of instructing modified tactics to crews which are otherwise fully qualified.

AN EXAMPLE

The most common of the tasks that are part of an AH-64 pilot's repertoire of abilities is a simple landing, called a VMC (Visual Meteorological Conditions) approach (Figure 1). The conduct of a VMC approach in certain environmental conditions is a relatively straightforward maneuver:

1. Determine an approach angle of approximately 8 to 10 degrees.
2. Decrease altitude and airspeed simultaneously to arrive at the touchdown point with little or no rate of descent or forward airspeed.

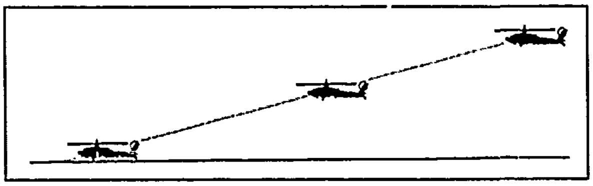


Figure 1 VMC Approach

As our aircrews began operating in the sands of Saudi Arabia during Operation Desert Shield/Storm they quickly discovered that performing this maneuver as they had been taught and trained caused a phenomenon known as brownout. Brownout is the creation of a large dust cloud as airspeed is slowed and the aircraft nears the ground (Figure 2). This dust cloud may obscure all contact with outside references and poses a potentially hazardous situation. During the training prior to Desert Storm, a number of rotary-wing aircraft were lost or destroyed as a result of brownout.

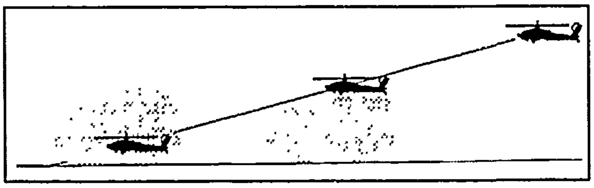


Figure 2 Brownout

Since none of the training systems currently available present this phenomenon, crews had to learn procedures to avoid the problem during actual aircraft operations. During Desert Shield/Storm aviators learned and practiced procedures that minimized the effects of brownout. One technique (Figure 3) involves attaining a much more shallow approach angle and maintaining a faster airspeed to put the dust cloud behind the aircraft.

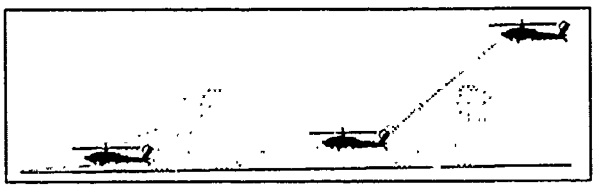


Figure 3 Brownout Avoidance Technique

This example illustrates very poignantly that although the crews were by definition fully combat ready, the training they received did not fully prepare then for operations in a desert environment.

Rapid turnaround simulation technology is one of the issues that has faced the designers of mission rehearsal systems. Mission rehearsal by definition is the training conducted in preparation for a specific mission. It also assumes that those participating in the rehearsal and subsequent mission are already at a high state of proficiency for those tasks, like a VMC approach, necessary for the accomplishment of the mission. Mission rehearsals are then conducted to coordinate those previously proficient tasks into mission accomplishment. Sustainment training is the process of learning and maintaining proficiency in those tasks necessary for mission accomplishment.

THE NEED FOR QUICK RESPONSES

Since 1979, the U.S. has been involved in six major military contingency operations: Operation Eagle Claw (1979-80), Operation Urgent Fury (1983), Operation El Dorado Canyon (1986), Operation Earnest Will (1987), Operation Just Cause (1989), and Operation Desert Storm/Shield (1990-91). Also during this period of time, the British Armed Services became engaged in a conflict with Argentina over control of the Falklands and South Georgia Islands (1982). Each conflict resulted in a quick-reaction contingency operation, with Operations Eagle Claw and Desert Storm having the longest planning to execution cycle. Eagle Claw was planned, practiced, and executed in a period of 172 days,[4] while execution of Desert Storm occurred 171 days after the invasion of Kuwait by Iraq. Urgent Fury, El Dorado Canyon, Earnest Will, Just Cause, and the Falklands conflict were all quick-reaction contingency operations which were planned and executed in a much shorter time period. Existing training doctrine did not fully support the execution of these missions under the conditions in which they were carried out. Furthermore, the sustainment training capabilities of forces during these conflicts, for the most part, continued to enhance crew skills for a conflict scenario which was somewhat unrelated to the conflict at hand.

SIMULATION AND QUICK RESPONSE

Quick-reaction contingency operations rarely involve changes to the operational doctrine or tactics of an armed force. Since training requirements are derived from this doctrine, a quick-reaction change to training syllabi rarely occurs. Therefore, the simulation training community finds itself essentially unable to respond to short-term, quick-reaction situa tions. In a world undergoing rapid politica economic, and social changes, the failure to reac can be fatal.

When the terms "quick-response" and "simula tion" are used together, they generally refer to mis sion rehearsal. Mission rehearsal, though, is onl one element of mission training. Wiggers, et al.[5] re fer to a hierarchy of mission training including mis sion preparation, mission preview, and combat mis sion training, in addition to mission rehearsa Monette, et al.[6] expand these categories to includ traditional "school-house" training as well as ad vanced graduate level (continuation) training Quick-response modification is applied solely to graduate level training, since we can assume tha all crews participating in a contingency operatio are qualified at the graduate level. The issue tha remains is how to prioritize which modification should take place in the limited time period availabl in contingency operations.

Courtice[7] has divided the combat tasks of th warrior into three categories: his ability to accuratel perceive all essential elements of the combat envi ronment, his ability to make accurate decisions, an his ability to make decisions in a timely manner. An quick-response simulation must support thes three categories of combat tasks to be effective Miller[8] suggests the application of tactical signif cance when analyzing the requirements for trainin mission-ready crews. Tactical significance is th degree of importance an environmental event o condition has upon tactical decisions that are made Tactical significance was the tool which was applie when defining requirements for our case exampl of quick-response modification: Project Dese STAARS.

DESERT STAARS: A CASE EXAMPLE

Early in Operation Desert Shield, it became ob vious that a means of providing ab initio and sustai ment training to crews located in the Persian Gu was necessary. Additionally, the need to continu to prepare crews for Middle Eastern conditions pri to deployment was also required. The Desert Sus tainment Training for Army Aviation Readines through Simulation (Desert STAARS) program wa intended to provide both of these capabilities fo U.S. Army Aviation crews.[9] Implementing a quick response modification program such as Dese STAARS is more complex than is apparent at co ceptualization. Since many of the tasks to b trained have, in fact, rarely been performed, definin the training requirements is, at best, a dynamic pr cess. The dynamic nature of training requirement is brought about by an unclear understanding of th

requirements by both the user and system designer at the time of conceptualization. High-level concepts of a *potential* plan do not uncover the necessary intricate detail of a plan which is often exposed as the plan is implemented.

As AH-64 crews were being fielded in Southwest Asia, the immediate need was to provide a system that would allow AH-64 crews to perform systems employment sustainment training. These switchology and procedural skills were determined to be the most volatile due to a somewhat limited availability of training assets and environments.

A major training and operational obstacle confronting allied forces upon their arrival and subsequent training in Southwest Asia was an inability to visualize the area where the fighting was expected to occur. Desert flying presents aircrews with phenomena that are not encountered during normal training in either the aircraft or the training devices. Extremely high ambient air temperatures, blowing sand and dust, desert haze, and the effects of rotor downwash on terrain (brownout) were identified as crucial environmental conditions that had tactical significance and were therefore required as part of Desert STAARS. The lack of detailed maps (some crews were reported to have trained using tourist road maps) prevented even a cursory topographical inspection of the battle area.

Geospecific visual data base technologies have become a major point of interest in the training industry. Although the technology is not fully developed, it was decided that a Kuwait Theater of Operations (KTO) data base would be built. The KTO data base provided crews the ability to visualize a potential battle area, thus providing mission-similar training. Once the KTO data base was decided upon, it became clear that further enhancements would not only prepare crews for the theater of operations, but might also provide mission-specific, and potentially mission rehearsal, capabilities. This presumes that Desert STAARS was afforded access to the latest Desert Storm mission plans. Since Desert Storm was still in the early planning stages when Desert STAARS was conceptualized (December 1990 I/ITSC Conference), and since the plans were highly sensitive at the time of Desert STAARS contract award (January 14, 1991), a complete set of mission specific locations was not incorporated and mission rehearsal was not provided in the Desert STAARS baseline.

The existing CMS design allows target vehicles to be positioned only at certain predefined, unchangeable locations. The ability to present actual or realistic threat/friendly vehicle posturing drove the requirement for target site relocation capabilities. This enhancement allows crews during training to engage known, suspected, or hypothetical formations at desired locations.

New threats were added to complement the existing library by providing vehicles which would likely be encountered by allied forces operating in the KTO.

It was also felt that the ability of crews to interact with an integrated threat force, including command, control, communications, and intelligence (C^3I) reporting chains and procedures, had tactical significance.

A sustainment training capability has little value if the personnel for whom it was intended are unable to get to the training. Desert STAARS originally intended to deploy training to fielded units participating in Desert Shield. Early in the program, it became apparent that deploying a complex, full-fidelity Combat Mission Simulator a distance of over 3,000 miles into possibly hostile territory would not be easy. The deployment issue had the potential to seriously impact the Desert STAARS development schedule, if not considered separately. The effort was divided into two operations: an engineering design and development effort and a simultaneous (or subsequent, if conditions warranted) deployment.

As Desert Shield progressed, it became apparent that the device might never be deployed. This raised the question: How do we use a system for sustainment training if we can't get the device to the users? The obvious answer: Bring the users to the device. A number of alternate sites were considered, including locations in Southwest Asia, Europe, and the continental U.S.

CONCLUSIONS

The radical changes in the world in the last two years, and their implications for U.S. military training and doctrine, have underlined the need to be able to respond quickly to international incidents and crises. U.S. military doctrine is based upon a concept of a threat which is rapidly becoming obsolete. There are no firm rules describing the methods of fighting against a new, and fundamentally different, political/military structure, as is currently evolving in Eastern Europe. How, then, can a specific training requirement be defined, especially for a short-term basis?

The invasion of Kuwait, which precipitated Operation Desert Shield, created a requirement for quick response from the training industry. Quick-response training is dependent on the accuracy of the preliminary mission requirements available. At the

conceptualization of Desert STAARS, we realized that we were faced with this dilemma as we tried to translate preliminary operational requirements into a set of training requirements.

Although the speed with which Operation Desert Storm terminated overshadowed the need for a sustainment training capability and the immediate use of the Desert STAARS project, several recommendations can be made from the lessons learned during this development.

First, quick-response modification of training systems is possible. The Desert STAARS contract called for a 90-day development and implementation time period. Desert STAARS was developed and operational on a CMS on the 90th day. Quick-response modifications to fielded devices suggest the need for a development device. Desert STAARS was fortunate to have the AH-64 CMS No. 1 device in-house and operational at the CAE-Link facility in Binghamton, New York, to support around-the-clock development during the very tight program schedule. It is necessary for a joint commitment between contractor and government to provide a dedicated device, even on a part-time basis, for quick-response modification. For future contracts, it is necessary to design in system flexibility.

The user must identify requirements quickly in order to meet short turnaround times. Distinction must be made between user needs and user desires and then the list must be prioritized based upon funds and time available. Industry must at the same time keep the government abreast of current technological limits and trends.

Efforts must be made to publicize the quick-response modification concept. With knowledge of the concepts and their availability, military planners can include training systems in their operational and logistical plans.

There are currently several interoperability initiatives under way in the training community. For the rapid modifications which were required for Desert STAARS, significant time could have been saved if these interoperability initiatives had been completed. The visual data base was constructed from available DMA data and then enhanced through the use of scanned photographs and drawings. A library of visual data bases, such as the one proposed for Project 2851, would have been immensely helpful in reducing the data base development timeline. It should be noted that Project 2851 did develop a KTO data base, but it was completed well after it was required for Desert STAARS. A fair amount of effort went into developing the networking interface between the CMS, FLS, and MULTISIM nodes. The MULTISIM baseline accounted for many of the data requirements of Desert STAARS, but had not accounted for geospecific network information, such as the existence of blowing sand from rotor downwash. The DIS standards, if completed and available, might have reduced the amount of time required to develop the network interface.

Quick-response modification implies the need for flexibility of design. Desert STAARS provided training aspects with the flexibility to train missions for which it was specifically intended as well as providing a quick response to an entirely new mission requirement. Incorporating quick-response modification to a training system, and positioning the system so that crews would have access, not only provides a method to sustain basic "switchology" and procedural skills but also provides a capability to train in an environment that

1. Depicts real-world conditions
2. Allows the safe conduct of training
3. Cuts down on training costs
4. Improves the overall quality of training

Based upon initial reactions of crews returning from the Persian Gulf to a demonstration of the Desert STAARS project, the application of quick-response modification to simulation can provide the deployed or deploying unit an unparalleled ability to improve and sustain combat skills. The training community needs to continue pursuing quick-response modification capabilities so that fielded units can, in fact, "train like we fight."

REFERENCES

1. Siegel, A.B., "U.S. Navy Crisis Response Activity 1946- 1989, Preliminary Report", Center for Naval Analysis, Alexandria, VA, November 1989.
2. Vuono,C.E., "Desert Storm, Aviation, and the Army of Tomorrow", Army Aviation, Volume 40, Number 6, May 1991.
3. U.S. Army Staff Officer's Handbook, RF101-999.
4. Brauer, R.F., "A Critical Examination of Planning Imperatives Applicable to Hostage rescue attempts", U.S. Army War College, Carlisle Barracks, PA, 16 April 1984.
5. Wiggers, R., Hiteshew, L.K., Matusof, R., "Mission Rehearsal: More Than Just Another Simulation", Interservice/Industry Training Systems Conference, Fort Worth, TX, November 1989.
6. Monette, R.J., George, G., Knight, S., "Mission Training and Rehearsal. Employing Simulation to Its Full Potential", Interservice/Industry Training Systems Conference, Orlando, FL, November 1988.

7. Courtice, A.J., "The Next Frontier of Air Force Training Technology", Interservice/Industry Training Systems Conference, Orlando, FL, November 1988.

8. Miller, D.M., "Foundations for Tactical Training: A Challenge to the Industry", AIAA Flight Simulation Technologies Conference, Boston, MA, September 1989.

9. Monette, R.J., Knight, S.K., Goodwin, P., "Sustainment Training for Army Aviation Readiness through Simulation", Interservice/Industry Training Systems Conference, Orlando, FL, December 1991.

ABOUT THE AUTHORS

CW3 Charles L. Fullmer is a Senior Army Aviator and an AH-64 Standardization Instructor Pilot assigned to D Company, 1-14 AVN REG, Fort Rucker, Alabama. With over 19 years of military experience, he has been involved with both air and ground training systems and programs. He has logged over 2500 hours in attack helicopters, including more than 1500 hours as an instructor in the AH-64 Apache Attack Helicopter. Currently he is a member of the Combat Mission Simulator upgrade and acceptance team, serving as an Attack Helicopter/Tactics/Threat Subject Matter Expert. CW3 Fullmer was one of two Subject Matter Experts devoted by the U.S. Army to the Desert STAARS program. CW3 Fullmer is completing a Bachelor's degree in Aviation Management with Embry Riddle Aeronautical University.

Victor Polkowski is a Systems Engineer with CAE-Link Corporation in Binghamton, New York. Mr. Polkowski has worked the last seven years in the simulation of avionics and tactical systems, primarily in the development of concurrency update programs to the AH-64 Combat Mission Simulator. Mr. Polkowski's most recent assignment was as the Systems Integration Engineer and Tactical Lead for the Desert STAARS program. Mr. Polkowski holds a Bachelor of Science in Computer Science from the State University of New York at Binghamton.

Ron Matusof is a Staff Engineer with CAE-Link Corporation in Binghamton, New York. He has most recently been involved in the integration of CAE-Link's MULTISIM Network for the U.S. Army Desert STAARS program. Mr. Matusof has in the last eight years worked in the simulation of avionics, tactical systems, and radar for a variety of military training devices and has spent over four years in the development and implementation of Link's MULTISIM simulator network. Mr. Matusof holds a Bachelor of Science in Electrical Engineering from the University of Pittsburgh. He has published several papers on the subjects of interoperability, mission rehearsal, and cue correlation.

TRAINING AND MISSION REHEARSAL FOR DEPLOYED NAVY AND MARINE AVIATION

CDR Paul Miles
Office of the Chief of Naval Operations
Washington, DC 20350-2000

Currently, a family of shore-based training devices is available to train flight crews to safely operate and fully employ their weapons systems. Weapons System Trainers (WSTs) are used extensively to provide mission training and to build and maintain proficiency. This capability is not available at forward-deployment locations for the Navy or Marines, and is not available aboard ship. Additionally, recent operations have disclosed the absolute need for deployed tactical aviation mission rehearsal capability. As a result, the Navy is pursuing the development of a family of training devices designed to serve remotely located tactical aviation units, under the overall program title of "Deployed Tactical Aircraft Training System (DTATS)."

"OK, kid Before you can fly, you've got to go down to the seventh deck and get your simulator hop. I'll see you there in fifteen minutes."

- long-standing Navy practical joke played on new carrier pilots

THE DEPLOYED TRAINING PROBLEM

In the early 1970s, Naval Aviation aircrew training was focused on preparation for two missions. prosecution of the air war in Vietnam, and strategic deterrence of the Soviet Union. At that time Navy and Marine Tactical Aircraft (TACAIR) were primarily the F-4, A-4, A-6 and A-7. Weapons for these aircraft consisted almost entirely of free-fall bombs, Shrike anti-radar missiles, and Sparrow and Sidewinder anti-air missiles. The delivery mode of these weapons was predominantly visual, using fixed sights and precalculated delivery parameters derived from weapons tables. The A-6 had introduced all-weather "system delivery" of ordnance, including "smart bombs," but actual use of sophisticated weapons was assigned to specially configured aircraft and specially trained crews; the bulk of A-6s continued to drop iron bombs. Meanwhile, the fighters also prepared for a visual war. "Fleet Air Defense" meant point-defense interception of a few threats by a few F-4s.

Today, we expect a squadron of F-14s to be able to repulse a regiment of bombers in the midst of sophisticated jamming, by using Phoenix missiles that can be launched in one of five different modes. We count on F/A-18 pilots to detect and destroy enemy aircraft, destroy enemy surface-to-air missile (SAM) and anti-aircraft artillery (AAA) sites with High-speed Antiradiation Missiles (HARMs), and deliver bombs onto pinpoint targets, all on the same mission -- at night. An A-6E crew must now be equally adept at all-weather, low-altitude attack; high altitude delivery of laser-guided weapons; War at Sea employment of Harpoon missiles and remote operation of Stand-off Land Attack Missiles (SLAMs).

While the TACAIR aircrew tasks of the 1970s were not as complex in comparison with today's, they often called for more individual skill and "hand/eye coordination." Attack pilots were required to physically aim their bombsights at their targets, skillfully noting and correcting for wind effects and deviations from precalculated parameters (such as dive angle). At that time, the Sidewinder was extremely simple to fire ("point, tone, shoot"), but achieving valid launch parameters with an F-4 against a non-cooperative adversary was not easy As a result of the need to build these *skills*, most mission training was conducted in actual flight, and involved repeated, skill honing practice. Training meant flying, performing practice intercepts, dropping practice ordnance and firing practice weaponry at practice targets.

Since then, we have worked hard in the design of our new weapons systems to make them easier to use. For example, the "Continuously Computed Impact Point" mode of the F/A-18 makes it much easier for a Hornet pilot to bomb accurately, and theoretically reduces his training requirement. However, the broader range of missions and the complexity of modern weapons and tactics have more than offset any potential training reductions. An F/A-18 pilot is now required to manage Forward-looking Infrared (FLIR) sensors, radar, HARM and advanced Sidewinders and [illegible] all on the same mission. Thus, in addition to piloting skills, an increased amount of weapons system experience must be developed.

The threat has advanced, too. Even if the Soviet Union should no longer be the Navy's main concern, modern Soviet weaponry still threatens us. The Navy and Marines confront a host of sophisticated weapons from sources all over the world, and some of our own that are in the hands of potential adversaries. In the early 1970s, the threat to TACAIR was AAA, SA-3 and SA-6 SAMs and MiG-21 jets with rear-hemisphere missiles. Today we must consider those same systems, as well as SA-10s, Crotales, Hawks, MiG-29s, Mirage 2000s, all-aspect missiles and sophisticated jammers.

Ashore, we have introduced some new training concepts to keep pace with these increasing training requirements. The effectiveness of our live flying program is greatly enhanced by the Tactical Air Combat Training System (TACTS), which has been integrated with Electronic Warfare (EW), SAM suppression and no-drop bomb scoring. But the biggest difference between shore training now and training "in the old days" is our current reliance upon simulators. Technological improvements in simulators now enable them to support training for most combat missions — a great advance from the basic instrument training and emergency drills of the early 1970s. Simulators have become the first introduction that young aviators receive to their fleet aircraft and weapons systems, and simulators continue to be relied upon for complex mission training after those aviators join the fleet.

Training while deployed is a somewhat different story, and is still almost entirely centered on flying. This means that during deployments, the classic training methods of the early 1970s are still the ones in use today. But the Navy's ability to conduct training in this manner has been steadily eroded by funding constraints. The flying hour program, which sought to provide an A-7 attack pilot with an average of 22.0 hours per month in 1983, will probably only achieve 20.8 hours average for an F/A-18 fighter *and* attack pilot in 1992.

Similar problems exist in providing the fleet with training ordnance, targets and adversaries, both at home and overseas. Training "space" is decreasing around the world, and is generally not sufficient in area or capability to practice tactics and employment the way we need to. We have a difficult problem replicating the complexity of the threat for training in the United States, much less overseas. Simulators have helped counteract these deficiencies at home, but we do not have an equivalent capability aboard ship or at our forward-deployment locations. Unfortunately, overseas is exactly where our training needs are highest.

The Navy is not about to willingly reduce its requirement to fly or to conduct live exercises with ordnance, targets and adversaries. We believe that we are right on the line with flying hours where further reductions will impact on safety. And because Naval Aviation depends so much on the skilled labors of our maintenance, flight deck, ordnance and engine room crews, flying is not just an *aircrew* training issue The problem is that our live training program is no longer sufficient to ensure the level of combat proficiency we want our deployed aircrews to maintain.

TACAIR MISSION REHEARSAL

Weapons system and threat improvements are not the only causes for increased deployed training requirements A combination of technological advance and military necessity has made "mission rehearsal" both feasible and in-demand. In the 1980s, Naval Aviation was presented with the task of frequently preparing, and sometimes executing, what became known as contingency operations (CONOPS) In many of these scenarios, aircrews were required to address heretofore unusual planning factors, such as the need to guarantee that no aircraft would be lost or that the strike would occur at a particular time of day, or wherein the "political message" was more important than actual destruction of the target. Many of these CONOPS plans

were reviewed at high levels Tactical surprise, defense suppression, target identification and first-pass delivery accuracy, always key elements of strike planning, became even more important. And unlike during Vietnam, the squadrons tasked with CONOPS rarely had any aircrews with previous experience over their potential targets. The Navy's requirements for mission rehearsal were born in this environment. While Desert Storm was not a classic CONOPS scenario, the potential lethality of Iraqi airspace, the timing of multiple missions and the desire to minimize collateral or friendly damage kept similar elements as major planning factors. Fortunately, early mission rehearsal systems (more properly "mission preview" systems) were available because of previous Persian Gulf operations (like "Earnest Will" and "Praying Mantis"), and their value has now been verified. As a result, the Navy and Marines recognize the need for mission rehearsal capability, and especially to have it on-scene with deployed forces.

CURRENT EFFORTS

The Navy has made several steps toward solving the deployed training problem and satisfying the need for mission rehearsal:

A-6E Systems/Weapons Improvement Program (SWIP) Part-task Trainer (PTT). Because shore-based trainers were not available for the A-6E SWIP aircraft in time for fleet introduction, the Navy developed this Ready Room trainer. The SWIP PTT, made by Delex Systems, combines low-fidelity flight simulation, sophisticated weapon and weapons system simulation and computer-based training. The SWIP PTT provides indoctrination and procedures practice for employment of the SWIP aircraft and the HARM, SLAM, Harpoon, and Maverick missiles, and is currently deployed with A-6E SWIP squadrons.

Networking. In 1990, the Navy and DARPA demonstrated the connection of the Combat Information Center of the USS Wasp to ship and helicopter simulators, with a simulated threat environment that represented a Mediterranean amphibious landing scenario Simulator Networking (SIMNET) protocols were used, and a secure link was demonstrated The Navy has signed a Memorandum of Understanding with DARPA to continue this development. The focus of current efforts is on supporting deployed and ashore tactical team training through networking, including the development of aviation networking standards, large-scale interactive threats, interaction between real and simulated systems and breakthrough visual display technology.

Embedded Training. The F-14A has had the capability for over fifteen years to generate simulated large-scale raids of fighters, bombers, missiles and EW on the cockpit displays while inflight. Aircrews maneuver as they would for actual threats, simulate firing missiles and receive a short debrief. For team training, one aircraft can send the scenario to wingman via E-2C datalink. The system can attach simulated jamming to actual non-jamming targets and can score simulated aerial gunnery against real targets Embedded training capability has become a requirement for future weapons systems.

Tactical Operational Preview Scene (TOPSCENE). In response to Persian Gulf operations of 1987 ("Earnest Will"), the Navy developed and fielded the first operational strike mission preview system. TOPSCENE generates real-time perspective images of target areas as they would appear from a moving aircraft, from overhead photography. The system, delivered in 1989 from LTV Corporation, presently consists of a database generation and mission preview system at the Naval Strike Warfare Center (NSWC) in Fallon, Nevada, and two carrier-deployable mission preview workstations. During Desert Storm, the two deployable workstations were in theater, aboard USS Theodore Roosevelt and with Marine Air Group 11. Additionally, NSWC produced videotapes of expected strike routes and target areas for the other ships and air groups that didn't have workstations. As a result of this wartime experience, mission preview is now an integral part of mission planning, and additional TOPSCENE systems are being procured.

Combat Training Systems. The various TACTS ranges have added significant value to our live training program through a combination of simulation, stimulation and networking. The TACTS range at NSWC can support 32 live aircraft, simulate SAMs and AAA, and provide detailed observation.

recording, debrief and analysis of an entire strike mission. Defense suppression, air combat and target attack are all supported. The benefit of TACTS to live training is so great that the Navy is funding a program to take TACTS to sea, as the Tactical Combat Training System (TCTS). TCTS will track aircraft, ships and submarines; stimulate radar and EW sensors; simulate large numbers of threats and provide engagement, recording and debrief for an entire Battle Group.

While each of these programs is important and successful, they are unable to solve the entire deployed training problem. And each has its limitations. For instance, in a time of fiscal austerity, keeping the A-6E SWIP PTT in the same configuration as the aircraft is becoming a challenge, and the training requirements of the A-6E SWIP community are beginning to outgrow the capability of the current PTT. Networking has great potential, but has a way to go technically, and is meeting cultural resistance — the Navy is not yet ready to force a large group of players to stop work and assemble for *simulated* operations. Embedded training systems can display sensor data, but can't present the visual aspects of the battle. We are examining on-deck, in-aircraft embedded training using helmet- mounted visual displays, but the development expense, aircraft wear-and-tear and loss of maintenance time are significant issues. Even the excellent training provided by TCTS will only be as good as the funding for our steaming and flying hour programs.

It is also important to note that TOPSCENE does not fully meet the Navy's requirements for mission rehearsal. TOPSCENE provides familiarity with portions of the mission environment (the appearance), but not with all of it. The Navy's mission rehearsal system must completely immerse the crewmembers into the same problems that they will face during the actual mission. We desire not just to teach them enroute navigation and target recognition, but to teach them to navigate while being opposed by the threat, and to recognize the target as part of the process of delivering a weapon. The threat, the cockpit, the weapons system and the weapons themselves must be a part of the rehearsal.

THE DEPLOYED TRAINING SOLUTION — DTATS

What is needed is a system that fills the gaps in our training program, that complements what we can do in the actual aircraft with systems like TCTS, and that meets the requirement for mission rehearsal. One approach is to grow the well-regarded deployable PTT concept into a full-fledged WST, and equip it with a TOPSCENE-like image capability and a threat environment based upon the real world, and to place this device where it's most needed. The result would be a Carrier-based WST (CVWST) and a nearly identical Deployable WST (DWST). These would be interfaced with available mission planning and intelligence data systems. The Navy program to develop these devices and interfaces is known as the Deployable TACAIR Training System (DTATS).

The particular requirements for DTATS are based upon fleet inputs and a projection of the technology available in the middle-to-late 1990s. DTATS should:

- Support Navy and Marine TACAIR (F/A-18, F-14 A-6, AV-8, EA-6 and AX) with a single reconfigurable hardware suite. This is to minimize the requirement for several different cockpits aboard ship and to reduce unit costs.
- Provide weapons delivery training for all strike weapons (air-to-air and air-to-ground) including classified weapons. This is to reduce the impact of not having enough captive and training ordnance.
- Provide mission training to include Interdiction, Close Air Support, Special Weapons Delivery, Defense Suppression, Fleet Air Superiority, Combat Air Patrol, Strike Escort, Tactical Reconnaissance, War at Sea and Minelaying. Each of these missions is difficult to practice realistically overseas. DTATS must complement TCTS by including the visual and electro-optical sensor aspects of these missions, such as overland navigation and target recognition, and by providing continuous availability when airspace is limited or flying is curtailed.

- Display realistic and correlated photography-based visual, radar and electro-optical sensor images, as appropriate to the aircraft and weapons being supported. A capability for limited on-site update of databases from the latest intelligence imagery is required.
- Interface with mission planning systems in a manner that simulates the appropriate aircraft's system, and with other available data systems such that actual threat order-of-battle can be simulated.
- Present realistic, responsive and intelligent threats derived from actual order-of-battle.
- Provide multi-crew and multi-aircraft training by connectivity with other simulators, DTATS devices and TCTS.
- Be compatible with the shipboard and forward-deployed environment in size, sturdiness and power and cooling requirements. DWST versions should be self-contained, not requiring special buildings for housing.
- Not require an operator or instructor, or dedicated maintenance personnel. The device should be simple enough to initialize and calibrate that trainees can do it themselves. The device should be self-diagnostic, and require only the infrequent removal and replacement of failed components by onboard military technicians. The goal is for a device that will operate for over 150 hours mean time between failures, and require less than 0.05 maintenance man-hours per operating hour.
- Be low enough in unit cost that sufficient numbers of devices can be bought. The goal is to procure at least one DTATS per aircraft carrier and amphibious assault ship, along with additional units for forward air bases, weapons schools and reserve squadrons.

ISSUES

Meeting all of these requirements will certainly push the simulator state of the art. In particular, the mission training requirement means that a wide field-of-view visual system will be needed. But certainly, the toughest requirement is to meet the packaging constraints imposed by carrier basing — the wide field-of-view visual display will be confined to a very small space.

The current concept of the CVWST version of DTATS is that it will be installed in a van-type enclosure on the ship's hangar bay. In the late 1990s, some EA-6B avionics maintenance vans in the hangar bay should no longer be required. This configuration would force DTATS into a package of about 8 x 8 x 24 feet. Such a configuration is probably also satisfactory for the DWST version, which would then be somewhat mobile and could be secured to a concrete slab without need for a building.

The reconfigurable cockpit approach to DTATS is also an issue, primarily in attempting to meet the needs of tandem cockpit aircraft like the F-14 and F/A-18D along with side-by-side configurations like the A-6. Three training stations might be required to satisfy both configurations. The type of visual display, whether helmet-mounted, virtual, mini-dome or whatever, certainly affects what layout is acceptable. For instance, if helmet-mounted displays are used, then a two-place side-by side layout with a removable divider might also be satisfactory for tandem cockpit training. The EA-6B, with its four-man crew also presents a problem, although it may be possible to accomplish the desired mission training by training only two or three crewmembers at a time. The "glass cockpit" approach, which uses a video monitor with touch screen as the instrument panel (perhaps with removable custom faceplates and side consoles for the different aircraft), appears to be a satisfactory.

The list of some of the technical and operational issues surrounding DTATS will challenge the Navy and those contractors that seek to build it.

- Visual display and image generator performance requirements
- Database size, source material, and production and update rate requirements
- Interface with mission planning and intelligence data systems
- Maintaining a match of configuration between the device and multiple aircraft and weapons

- Security requirements
- Requirements for Artificial Intelligence in scenario preparation, threat interaction and system operation and maintenance
- Detailed requirements for reliability, maintainability and suitability

A question asked frequently by industry is: "How much fidelity is required?" The actual answer (in specification language) is still somewhat undetermined. But as a guide, we will accept lower fidelity *within* the cockpit (i.e., panels, switches, g-seat, etc.), but demand greater fidelity *outside* the cockpit (visual display, threats, sensors and weapons).

R&D REQUIREMENTS

Obviously, a robust research and development (R&D) effort will be required to resolve these issues and achieve the capabilities that the fleet needs. The Navy has funded a small 6.3 R&D effort to support the CVWST since 1990. This effort is expanded significantly in 1992. In 1993, an Advanced Technology Demonstration is funded to specifically address the visual display issue. For the future, the Deputy Chief of Naval Operations (Air Warfare), OP-05, is committed to the DTATS program and intends to conduct a competitive DemonstrationValidation in 1994, followed by a first article contract award in 1995. The first DTATS/ CVWST would go to sea in 1998.

Of course there's no guarantee in today's budget climate that DTATS will ever be built. But the need is there, and sooner or later, we tend to put money where the need is. Meanwhile, it is encouraging to see the simulator industry already addressing, through in-house R&D, many of the technologies critical to DTATS The developments that the Navy has specifically identified so far include the following:

- Low cost, multichannel, photography-based image generator, capable of providing visual, sensor, radar and seeker images.
- Common, photography-based database for visual, electro-optical and radar displays, produced directly from multi-spectral overhead stereo imagery without human intervention.
- Low cost, wide angle, large field-of-view visual display system, with resolution approaching the eye limit, fitting within strict space limitations.
- Threat generator featuring realistic interactive threats, with order-of-battle imported from real-world Navy intelligence databases.
- Sophisticated simulator which can be operated by the personnel receiving training, capable of self-initialization, security monitoring, easy scenario selection and performance monitoring.
- Sophisticated simulator which is hardened for the carrier environment, highly reliable, self-diagnostic and self-calibrating; designed such that in the event of failure, deployed maintenance requires only removal and replacement of self-diagnosed components.
- Simulator architecture in which the configurations of multiple tactical aircraft can be kept accurate and current.
- An order-of-magnitude reduction in simulator footprint and unit cost.

The capabilities that are envisioned for DTATS will be in demand for all of our training devices. low cost, modularity, commonality, small footprint and mission rehearsal capability. Therefore, whether or not DTATS itself makes it aboard ship, the vendor that has developed technologies for DTATS will find a customer.

CONCLUSION

Other difficult issues and R&D requirements will certainly arise as the Navy attempts to place the first DTATS aboard ship in 1998. The most important immediate hurdle to jump is that of user acceptance. As the introduction to this paper shows, the desirability of a simulator aboard ship is not yet universally recognized, and the idea is not always taken seriously. Often what is taken seriously is that a device of this capability might pose a severe threat to the flying hour program.

There is a level of flying that we cannot safely go below — and the Navy is there already. For the DTATS concept to be accepted, it must not compete in purpose with what we can do in the air. DTATS must not attempt to teach the basics, and it

must avoid those things that simulators are generally regarded as doing poorly (such as carrier landing training). DTATS must be focused on things that can't be done with the real aircraft: complex scenarios, complex tactics, training for rare and complex weapons and, most importantly, full-fledged mission rehearsal. And the fleet aviator must be willing to utilize DTATS for it to be of any value. On its own merits, the device must draw aviators away from their collateral duties and out of their staterooms, or excuses to avoid it will be found. Therefore, DTATS must be challenging and fun: the best videogame on the ship.

Then, we'll need a new practical joke.

ABOUT THE AUTHOR

CDR Paul Miles was the Tactical Aircraft Simulator Requirements Officer for the Chief of Naval Operations at the time of this writing. He has a Bachelor of Aerospace Engineering degree from Georgia Tech, and has served 17 years in the Navy, primarily as an F-14 pilot. He is currently assigned as the Air Operations Officer aboard USS John F. Kennedy (CV-67).

VIRTUAL REALITY: A PRIMER
A DISCUSSION OF DEFINITIONS AND POSSIBLE APPLICATIONS FOR MILITARY TRAINING SYSTEMS

Dr. George C. Barcus
NAVTRASYSCEN
Orlando, Florida

Mrs. Theresa T. Barcus
NAVTRASYSCEN
Orlando, Florida

Mr. Richard R. Dunn-Roberts
Institute for Simulation and Training Technology
Orlando, Florida

Abstract

This paper reviews the latest developments in and traces the background of the "new" technology of Virtual Reality. Concepts covered will include the AIP cube, physical and geometric modeling, dead reckoning and behavioral modeling. Intended as a primer, through this article the reader will be introduced to the field of Virtual Reality by explaining common terms, theoretical concepts, enabling technologies and by presenting present and future applications of virtual environments.

Introduction

In the last few years, a significant amount of attention has been directed towards interactive simulations that take advantage of technical developments in human-computer interface technology. The popular name used to describe this type of simulation is Virtual Reality (VR). Other names include virtual environments and virtual interfaces. While advocates of the interface model claim extensive and far-reaching applicability for the paradigm, the current state of the technology is crude and intrusive. In spite of this, there is significant excitement about VR in academic research establishments, our military training communities, and the commercial entertainment industry.[1]

Background

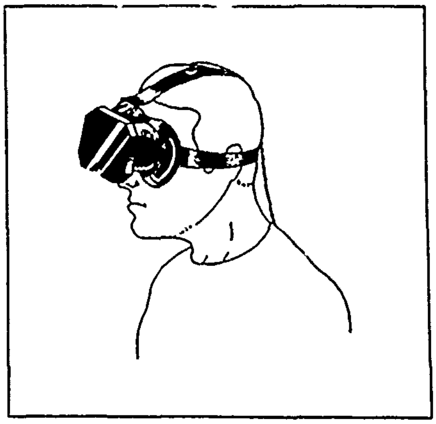
The term "virtual reality" has been described as a collective term for a family of computer technologies that are capable of generating apparent three-dimensional space (referred to as cyberspace) where a person can achieve the "sense" of moving around and doing things within this space. Computer graphics visionary Ivan Sutherland first introduced the concept of virtual reality within the computer in 1965 when he described a computer monitor screen as a window through which one sees a virtual world. Even earlier, in his doctoral thesis, "Sketchpad", Sutherland described the first complete system for making drawings with a computer. Virtual reality has existed for over twenty years as a research tool first used by the Air Force in jet flight simulation and by NASA in preparatio for the Apollo moon missions. The technology exists today and is being advanced by many university computer science departments, government and civilian research labs, and military projects.[2] It also exists in the field of computer-aided design (CAD), in which a designer and end user can observe, explore, and manipulate computer-generated objects. To a very limited degree virtual reality also exists in the entertainment world in the form of "space" games. As an educational tool, VR has been used to bring the student within a model or conceptual environment to be studied to better explain its content, structure or purpose.

While at the University of Utah in 1968, Sutherland developed a 3-D display system that reacted to the user's head movements. This system provided a wrap-around virtual environment. In 1974, Myron Kruger, then a graduate student at the University of Wisconsin at Madison and now a Connecticut computer scientist, coined the phrase "artificial reality" while conducting research and experiments to test his belief that humanity's relationship to technology can be a positive experience. Presently, one of his environments, VIDEOPLACE, installed at the Connecticut Museum of Natural History in Storrs, Connecticut, continues to be used to further this research in man-machine interaction.[3]

The Department of Defense in 1978 invested over a million dollars to develop a 3-D display simulator as part of the project called Visually-Coupled Airborne Space Simulator (VCASS) for pilot training. The outgrowth o. this work eventually produced the Super Cockpit. This was one of the earliest military applications of

virtual reality and arose from work carried out at the Wright Patterson Air Force Base. The goal of this work was to develop advanced avionics and cockpit management systems to permit the screening of pilots of future military aircraft from direct visual contact with the outside world.[4]

In 1984, the Human Interface Research Laboratory at NASA's Ames Research Center developed the first lightweight stereoscopic head mounted display based on miniature Liquid Crystal Display (LCD) monitors, and funded a small California computer company, Virtual Programming Languages (VPL) Research, Inc., for development of the DataGlove. [5] The DataGlove consists of a glove fitted on the back with a network of fiber-optic cables connected to a light-emitting diode at one end and an array of photosensors at the other. Flexing a finger reduces the flow of light where a fiber-optic cable is bent at a knuckle or joint; the resulting change in light intensity is translated into positioning data that can be transmitted in digital form to a computer. The most common version of this type of input device, although admittedly crude compared to its possible potential, is Mattel Inc.'s Power Glove.[6] This is a joystick substitute used for controlling Nintendo games. Subsequent research has produced VPL EyePhone goggles which is a display mechanism comprised of two small color LCD video monitors in a counter balanced head-mount. An example of this type of system is shown in Figure 1. Wide angle optics in front of the eyes give the user an approximately 120 x 60 degree field of view. This system uses two National Television System Committee (NTSC) composite video signals. Presently, VPL Research Inc. founder, Jaron Lanier is directing the development of an expermental computer network, known as RealityNet, that will attempt to make virtual reality transmittable over standard telephone lines. Partners in this system will include the University of Washington Human Interface Technology Laboratory, the Computer Science Department at the University of North Carolina, and the Massachusetts Institute of Technology (MIT) Laboratory in Cambridge, Massachusetts.



Head mounted virtual reality display mechanism

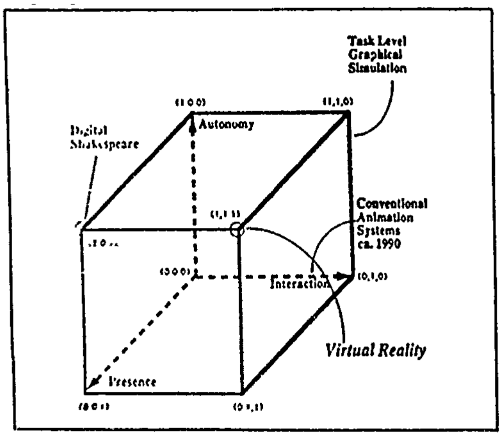
FIGURE 1

International interest in the field of virtual reality is evidenced by the work of the Advanced Telecommunications Research Institute, a consortium of 150 firms, mostly Japanese, which are devoting in excess of 5 million dollars per year for ten years to explore "communications with realistic sensation." A related area of research comprising virtual reality is involved with tactile feedback. The TELETACT Virtual Tactile Feedback System (a registered Trade Mark name of Airmuscle Limited manufactured under licence by the Advanced Robotics Research Limited Company) was designed to investigate the lack of tactile and force feedback when interacting with objects in a virtual environment. By using a glove-based stimulation device, this system assists in permitting experimenters to generate simple tactile and force patterns for a variety of objects. These can be stored on computers and recalled to identify "contact" with virtual objects.[7] Current efforts in virtual reality include work at the Aerospace Human Factors Research Division of NASA's Ames Research Center where an interactive Virtual Interface Environment Workstation (VIEW) has been developed to aid design, simulation and evaluation of advanced data display and management concepts for operator interface design. The VIEW system will provide a virtual auditory and stereoscopic image surround that is responsive to inputs from the operator's position, voice and gestures. As a low-cost, multipurpose Input/Output (I/O) device, this variable interface configuration will allow the operator to virtually explore a 360 degree synthesized environment. It can also be used to remotely explore hazardous environments or control robotic devices at a distance.[8]

Efforts known as Telerobotics and Telepresence are examples of VR technology that help remove man from hazardous environments. The Advanced Robotics Research Limited Human Factors Program called Virtual Environment Remote Driving EXperiment (VERDEX) addresses the design of advanced human system interfaces which permit intuitive interactions between a human operator, a robotic vehicle and a remote environment. VPL Research, Inc. has recently developed a system that allows more than one user to share a virtual space. Known as Reality Built for Two (RB2), it is a development platform for designing and implementing real-time virtual realities.[9] Additional VR interests involve 3-D sound, advanced development platforms and improved tracking mechanisms such as the Polhemus Isotrak.[10]

In spite of the flood of development of virtual reality systems, VR is in serious danger of being oversold by the media. To understand the problems facing VR developers we will look at some theoretical work and concepts arising from that work. The AIP cube proposed by David Zeltzer of MIT's Media Laboratory is a qualitative tool for evaluating virtual realities. The AIP cube, as displayed in Figure 2, provides for a taxonomy of virtual environments based on three components Zeltzer considers to be salient. These components are autonomy, interaction, and presence, hence the name of the cube.

Each of these components represent an axis of the cube, and can be considered to have a value between zero and one. The autonomy axis represents the sophistication of the computational model underlying the simulation. If there is no model, we have a static geometric model that can be rendered as a picture, but has no capability for autonomous behavior. This is represented as an autonomy value of zero. An autonomy value of one would indicate a very sophisticated model, supporting a high level of autonomous and emergent behavior in the simulation. The interaction axis represents our ability to affect the simulation during the runtime cycle. An interaction value of zero would be indicative of a graphical simulation that could not be affected once started. An interaction value of one represents allowing the user to modify any and all model parameters during runtime. It is not desirable, however, to require the user to maintain control over all model parameters, because the user could easily be overwhelmed.



AIP CUBE: Autonomy, interaction and presence

FIGURE 2

The final axis of the AIP cube is the presence axis. This refers to the degree to which the user actually feels immersed in the virtual reality. Presence is supported by the I/O peripherals (such as a Helmet Mounted Display (HMD)) through which the user perceives the VR. This can also be expressed as the success with which the I/O channels of the virtual reality map into the I/O channels, or senses, of the human user. Again, a value of 0 represents the least amount of presence, such as a keyboard and text based monochrome monitor, while a value of 1 would represent a VR in which the user feels completely immersed.

Modeling, introduced above under autonomy, is an area of great interest in VR research. Modeling can be broken down into three areas: geometric modeling, physical modeling, and behavioral modeling. The geometric model represents what the object being modeled looks like ... its shape, color, texture, etc. The physical model represents the mechanics of the object's behavior. The behavioral model represents higher level abstract behaviors.

Definitions

Before continuing, it may be helpful to review and explain a few of the more common terms relating to the field of virtual reality. Although presently definitions are not stable, many being coined or trademarked as new applications and products are discovered and produced, quite a few have attained "common usage" status. A sampling of these include:

- Convolvotron
  A signal processor that coordinates a target location and the position of the listener's head and "places" the signal in the perceptual 3-D space of the user.
- Cyberspace
  While there is more than one definition listed, for this paper, the first definition more closely describes our use of the term "cyberspace".
  -(John Walker of Autodesk) Cyberspace is a system that provides the user a three-dimensional interaction experience that provides the illusion he is inside a world rather than observing an image. At the minimum, a cyberspace system provides stereoscopic imagery of three-dimensional objects, sensing the user's head position and rapidly updating the perceived scene. In addition, a cyberspace system provides a means of interacting with simulated objects.
  -(William Gibson) First coined the term cyberspace in the 1982 trilogy NEUROMANCER, COUNT ZERO, and MONA LISA OVERDRIVE. This referred to a 3-D representation or model of information. This representation could be used to access or modify the underlying data. Cyberspace "tricks" participants into believing they are within a place, real or imaginary, apart from their location in physical space.
- Cybernetics
  The science of communication and control theory that is concerned especially with the comparative study of automatic control systems (as in the nervous system and brain and mechanical-electrical communication systems).
- Distributed Virtual Reality
  A VR program running on a distributed network of computers.
- Virtual Reality
  While all definitions listed are correct, Mr. Dunn-Roberts' definition is the one the authors used in this paper.
  - (Dunn-Roberts) Denotes a multi-sensory real-time simulation that immerses the user in 3-D graphical space through the use of innovative I/O (Input/Output) technology, such as head-mounted displays. VR allows freedom of movement within the space, and supports complex interactions including the modification of features of the space itself. Other terms used are artificial reality, cyberspace, virtual environment and virtual world.
  - (Emery) Virtual reality is described as a collective term for a family of computer programs that are capable of

generating apparent 3-D space where a person can achieve the "sense" of moving around and doing things within this space.

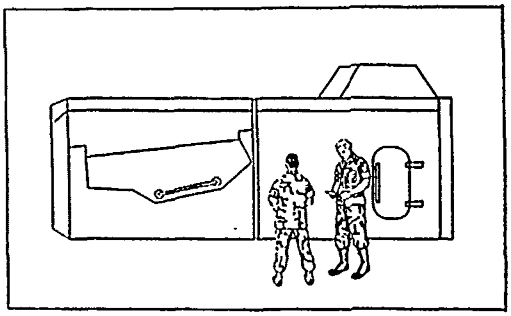
- (Stone) Virtual Reality broadly refers to the generation, using computer graphics, of realistic three-dimensional visual, audio and tactile worlds in which a suitably-equipped user can explore and interact with virtual objects using natural human skills.

Present Military Applications

Based on Zeltzer's taxonomy, no current systems are complete virtual realities. However, there are many research efforts that are moving us towards virtual reality. Many of these research efforts have been and are being conducted under the auspices of military research, although not always with the specific intent of furthering our virtual reality capabilities. As mentioned above, Tom Furness's VCASS project was one of the first military projects to use head-mounted displays in association with computer generated three-dimensional worlds. This system was developed to look at the use of advanced interface technologies to improve pilot situational awareness in fighter aircraft. Project scientists realized that a fighter pilot must operate his aircraft in a 3-dimensional space while the information necessary to that operation is presented to the pilot came from highly coded two-dimensional representations of the world. Often, in order to obtain that information, the pilot must lower his head from the real world in which he must operate. Furness wanted to take advantage of the human capacity for handling properly presented spatial information. The Super Cockpit used a helmet-mounted video display system to either overlay the real world with imagery, or to completely replace the real world with a synthetically generated representation of the real world. Information that had a spatial context for the pilot, such as threat information, was conveyed in spatially relevant positions. Three-dimensional audio also conveyed directional information. Both head and eye position were tracked to facilitate selection of virtual switches. Voice and gestural command input was also allowed.

However, the Super Cockpit is not the only military research project that has VR applicability. The Visual Technology Research Simulator Laboratory (VTRS) at the Naval Training Systems Center in Orlando, Florida, has worked on many aspects of simulators that may have future impact on VR systems. Projects done at VTRS looked at questions involving motion bases on training simulators and the use of head and eye-tracking equipment in dome based simulators.

SIMNET, a large scale networked team training simulation system, is not considered a virtual reality system; however, it is another important forerunner to a virtual reality system. SIMNET, as shown in Figure 3, actually provides many of the characteristics of a virtual environment and adds at least on important possible component of future VR systems. SIMNET provides a multi-sensory (sight and sound) immersion in a synthetic environment where users can interact with many other participants. While the user's windows into the virtual world are small, a shell representing an M1A1 Abrams tank helps improve the sense of presence. Some of the participants the user interacts with are other humans, as well as others being autonomous enemies controlled by a computer. One of the most important contributions made by SIMNET is the concept of dead reckoning. Dead reckoning allows each participant in the networked simulation to keep track of the location and state of all other participants while reducing the network traffic. Each participant stores location and velocity information for all other participants, and updates their locations during each time slice in the simulation. If a player's real location deviates from the location determined by dead reckoning, the player sends out an update to the other participants to correct their perception of the player.[11]



SIMNET: A large scale networked team training simulation system

FIGURE 3

However, as stated, SIMNET falls short of virtual reality. Interaction in SIMNET is limited to interactions between players. The database representing the world in which a SIMNET exercise takes place is static, and cannot be affected by the players. Also, presence is limited by the small size of the user's viewports into the world. In order to address these issues, the Army Project Manager for Training Devices (PMTRADE) has funded several research projects at the University of Central Florida's Institute for Simulation and Training (IST).

The first of these projects is looking at the integration of low cost head-mounted displays with several image generators, including SIMNET and an Evans and Sutherland ESIG-500. This project is looking at the feasibility of using head-tracking capability and HMD's to improve the user's access to the world modeled in training simulators, including head-out-of-the-hatch operation and dismounted infantry capabilities. This work explores technologies that may improve a trainee's sense of presence in the virtual world in the training simulator. [12]

Another project underway at IST is exploring the algorithmic requirements to increase the level of interaction with the virtual world at runtime in a training simulation. This project has modeled a virtual bulldozer that can dig trenches and

tank defilades, as well as a water flow model that simulates the flow of water in the virtual world. These capabilities are representative of physical models that are not currently available in real-time image generators, and will be necessary to provide interaction and autonomy in a virtual reality.

Also underway at IST is a project to develop a laboratory testbed for evaluating new VR technologies. The Virtual Environment Test Bed (VETB) is based on an IST developed networking protocol to support virtual environments running on a distributed heterogeneous network of computers.[13]

In addition to projects that are advancing the enabling technologies of virtual reality, some projects take advantage of VR technology to support other efforts. At NASA Ames Research Center, the Crew Station Research and Development Facility (CSRDF) is using a simulator to assess proposed designs for the LHX light attack helicopter. The simulator is built around a General Electric Compuscene 4 image generator driving a CAE Electronics HMD. This is an example of the use of virtual reality technology for design. The Army expects to save about $1 billion by not building prototypes of the LHX.[14]

The CSRDF simulator is similar to several other simulators based on CAE HMDs. These include an F16 pilot trainer at the Air Force Human Resource Laboratory at Williams Air Force Base and Army Research Institute Helicopter simulator at Fort Rucker.

Future Military Applications

Future applications of VR in support of training are limited only by how far we have progressed in moving systems along the three axes of Zeltzer's AIP cube. As stated earlier, the current state of the technology is not sufficient to support all of the applications we can imagine supporting. However, research into display technology, force feedback, computational modeling and algorithms is improving our capabilities. The virtual worlds we can build will improve.

Perhaps the most obvious use of virtual reality technology in support of training is the idea of a virtual simulator. In the morning, a team of army aviators might use the virtual simulator to practice flying attack missions just above tree level. After lunch, a crew of Navy aviators practice carrier landings. Later, a group of human factors scientists may test cockpit configurations. To support the virtual simulator and allow for different kinds of controls, improvements will be required in HMD resolution and tactile and force feedback capabilities.

Training simulators could be enhanced through networking to provide for large scale training exercises. They could also be enhanced through the use of autonomous enemy forces to provide training in strategy and tactics. VR technology may provide for the inclusion of dismounted infantry in the combined arms training exercise, although there are some difficulties associated with this. For a long-range view of the prospects for seamless integration of dismounted-infantry into combined arms simulations, see the work of D.K. McBride.[15]

An interesting variant for dismounted infantry training in VR involves the use of Personnel Amplification Systems (PAS), such as the ones studied at Los Alamos National Laboratory in the PITMAN project. This engineering study examined issues ranging from force feedback and servo design to visual displays. Such a suit could actually provide the interface through which a soldier would be immersed in the virtual world, complete with force feedback.[16][17]

Finally, we wish to look into the not so near future and expand on the possibilities in the use of VR for training commanders in strategy and tactics. We can imagine a virtual war college, where commanders could visit simulated battles that had either been fought or were under planning. In observing virtual battles, commanders could take advantage of the ability to scale time and space to gain a God's eye view of the battle while jumping through time to interesting moments in the course of the battle. The spatial correspondence of the virtual battlefield to the real battlefield would aid commanders in placing units on the virtual battlefield to test what-if hypotheses about battles. This virtual war college is far in the future, if it can be attained at all. However, this is the type of application that helps maintain the excitement that virtual reality is creating.

Conclusions

Although the material presented here is by no means exhaustive, the intent is to introduce the reader to the field of virtual reality and suggest a starting point for learning more about the subject. As VR applications and related computer technologies advance, the use of virtual reality and its associated techniques will become more common place in the future development of military training systems.

REFERENCES

[1] Dunn-Roberts, R. World One, Node One Prototyping a Virtual Reality Compiler. Master's Thesis Proposal, University of Central Florida, Spring 1991. [2] Ditlea, S. Welcome to Virtual Reality. Sourcebook, pp. 58-61, 1990.

[3] Kruger, M. Artificial Reality: Past and Future. Multimedia Review, pp. 31-34, Summer 1990.

[4] Furness, T. The Super Cockpit and Its Human Factors Challenges. Proceedings of the Human Factors Society 30th Annual Meeting, pp. 48-52, 1986.

[5] Ditlea, S. Another World: Inside Artificial Reality. PC/Computing, pp. 92-101, November, 1990.

[6] Bricken, M. Inventing Reality (Excerpts from an Autodesk paper, January 1989) CADalyst, pp. 45-46, December 1989.

[7] Stone, R.J. Virtual Reality: Interfaces for the 21st Century, Advanced Robotics Research Limited, Salford, UK, 1990.

[8] Fisher, S., Wenzel, E.M., Coler, C. and McGreevy, M.W. Virtual Interface Environment Workstations. Proceedings of the Human Factors Society 32nd Annual Meeting, pp.91-95, 1988.

[9] Blanchard, C., Burgess, S., Harvill, Y., Lanier, J., Lask, A., Oberman, M., and Teitel, M. Reality Built For Two: A Virtual Reality Tool, VPL Research Inc., Redwood City, California, 1990.

[10] Polhemus Isotrak is a trademark of Polhemus Systems, Inc.

[11] Moshell, J.M., Dunn-Roberts, R.R., and Goldiez, B. Virtual Reality for Military Training: SIMNET, Ender's Game and Beyond. Submitted for publication to Cyberspace, MIT Press, Cambridge, MA, May 1990.

[12] Dunn-Roberts, R.R., Da Silva, R. Altman, M. First Year Report, BAA Contract #0041, Task #1: Visual Display technology R & D, Institute for Simulation and Training, University of Central Florida, Orlando, FL, March 1991.

[13] Moshell, J.M. The Architecture of the Virtual Environment Testbed. VSL memo 91.6., Institute for Simulation and Training, University of Central Florida, Orlando, FL, April, 1991.

[14] Henderson, B.W. Simulators Play Key Role in LHX Contract Selection, Aviation Week & Space Technology, McGraw-Hill, November 27, 1989.

[15] McBride, D.K. Individual Portal (I-Port) in Advanced Distributed Simulation, Research, Proceedings of the ADPA Conference on Close Combat (Light) Simulation Technology, Albuquerque, NM, 17-20 September, 1990.

[16] Gorman, P.F. SuperTroop via I-Port: Distributed Simulation Technology for Combat Development and Training Development. IDA Paper P-2374, Institute for Defense Analysis, Alexandria, VA, August, 1990.

[17] Moshell, J.M., Smart, E.A., Dunn-Roberts, R.R., Blau, B.S. and Lisle, C.R. Virtual Reality: Its Potential Impact on Embedded Training. To appear in the NATO Workshop on Advanced Training Systems, Venice, Italy, October 22-24, 1991.

SUGGESTED READING

Emery, G. "Virtual Reality's Radical Vision." Insight on the News, pp. 20-25 May 6, 1991.

Fisher, S., Wenzel, E.M., Coler, C., MCGreevy, M. Virtual Interface Environment Workstations. Proceedings of the Human Factors Society, 32nd Annual Meeting, 1988.

Furness, T.A. The Super Cockpit and its Human Factors Challenges. Proceedings of the Human Factors Society 30th Annual Meeting, 1986, pp. 48-52.

Gibson, W. Neuromancer, New York: Ace Books, 1984.

Helsel, S.K., and Roth, J.P. Virtual Reality: Theory, Practice and Promise. Meckler (Westport/London), 1991.

Josephson, H. Virtual reality is virtually round the corner. New Scientist v.127 p. 30, September 22, 1990.

Polhemus Navigation Sciences Division. 3Space Isotrak User's Manual. McDonell Douglas Electronics Company, 1980.

Stone, R.J. "The Best of Both Worlds": A Combined Virtual-Real Human-Computer Interface for Telepresence and Remote Driving. In Proceedings of '91 ISART-International Symposium on Advanced Robotics Technology; Tokyo, Japan; March, 1991.

Stone, R.J. Virtual Reality in Telerobotics, Proceedings of the Computer Graphics '90 Conference, London; Blenheim Online, pp. 29-39, 1990(b).

Sutherland, I.E. "Sketchpad - A man-machine graphical Communication system", Proceedings of the Spring Joint Computer Conference. Detroit, Michigan, May 1963 and MIT Lincoln Laboratory Technical Report No. 296, January 1963.

Sutherland, I.E., "The Ultimate Display", Proceedings of the IPIP Congress 2, pp. 506-508, 1965.

Walker, J. Through the Looking Glass; Autodesk Inc. Sausalito, CA, September 1988.

Wenzel, E.M., Wightman, F.L. & Foster, S.H. A virtual display system for conveying three-dimensional acoustic information. Proceedings of the Human Factors Society, 32, 86-90, 1988(a).

Zeltzer, D. Virtual Environments: Where Are We Going? Proceedings of the 12th International IDATE (Institute de l'Audiovisual Telecommunications en Europe) Conference, Montpellier, France, November 14-16, 1990.

ABOUT THE AUTHORS

Dr. George C. Barcus is the Project Director for Advanced Technology within the Program Director, General Education Methodology and Technology at the Naval Training Systems Center, Orlando, Florida. He holds a Doctorate Degree from the University of Central Florida in Education, a Master's Degree from the Ohio State University in Instructional Media and an additional Master's Degree from the University of Florida in Curriculum and Instruction.

Mrs. Theresa T. Barcus is an Education Specialist in the Strategic Planning Group at the Naval Training Systems Center, Orlando, Florida. She holds a Master's Degree in Education from the University of Florida and is presently pursuing a Specialist Degree from the University of Central Florida.

Mr. Richard R. Dunn-Roberts is a Visual Systems Scientist at the Institute for Simulation and Training at the University of Central Florida, Orlando, Florida. He is currently working in the Visual Systems Laboratory on projects exploring the use of advanced display technology with real-time image generators and training simulators. He has a B.S. degree in Computer Science from the University of Central Florida, and for his Master's Degree is exploring design and integration requirements of virtual reality.

INVESTIGATING THE SUITABILITY OF SPEECH RECOGNITION FOR TRAINING SYSTEMS

Robert Rejent
Catherine Meyn

Logicon, Inc.
Tactical and Training Systems
San Diego, California

— ABSTRACT —

Speech recognition can promote enhanced training procedures and reduce operating costs in training systems. For this reason, the incorporation of speech technology into training systems is becoming more prevalent. Many users of these training systems, however, are unaware of the technical capabilities of speech recognition, and therefore have unrealistic expectations which affect trainer acceptability. To prevent this, it is important for the user and developer of any training system to probe the question. "Is speech recognition appropriate for this training application?" Logicon has integrated speech technology into air traffic control training systems for nearly 15 years. In transitioning from research and development systems to fully operational trainers, experience has been gained regarding this fundamental question. This paper identifies the issues associated with determining the suitability of speech recognition for a particular training application.

— BACKGROUND —

Speech recognition has been identified as a means for improving training capabilities and reducing operating costs. Because of this, requirements for state-of-the-art trainers are including speech technology to meet training needs. Those responsible for defining these requirements, however, are often unfamiliar with the capabilities of speech recognition. In addition, many potential users of training systems have unrealistic expectations of how the technology should perform. Much of their exposure to speech technology has been through television and movies. Robots like C-3P0 and R2D2 in "Star Wars" which understand all human speech might sell movie tickets, but do not represent the current capabilities of speech technology. The rapid advances in speech recognition devices also confuse the issue, as requirements analysts and system developers try to incorporate constantly changing capabilities.

Understanding the capabilities and limitations of speech technology permit the user to determine whether speech recognition is appropriate for a particular training application This is most appropriate when the requirements of the training system are identified. Collaboration between the user and developer is essential during this phase The user and developer must determine whether the incorporation of speech recognition into the trainer can support the training objectives given the current state of speech technology.

* *Logicon is primarily involved with speaker dependent, continuous speech recognition, which permits the most robust and natural speaking style. For this reason, the focus will be on this recognition type.*

— COMMON SPEECH RECOGNITION — ASSUMPTIONS

Although it is not necessary for the user to understand speech technology in an engineering sense, there should be some understanding of how it works in general.*This is necessary to avoid misconceptions and unrealistic expectations. For example, it has been common in our experience for the user to assume that the more commands defined for the recognition system, the better the chances of a command being recognized. In fact, the larger and more complex the phraseology, the more difficult it is for the recognition system to work effectively. As it happens, this is true for the student as well, so constraining the training phraseology to the minimum number of commands required to meet the training objectives meets the dual goal of improving recognition and reducing student training requirements.

Another assumption which is commonly made is that a command which is misrecognized is a failure on the part of the system and therefore a "bad" thing. It is important to remember that a speech recognition system is designed to recognize. It is also important to remember that a speech recognition system does not know of any words outside of the training phraseology. This means that the system will always identify a match with what was spoken. While these truths constrain the technology somewhat, if properly understood they can provide significant benefits for reinforcing training. In fact, poor recognition which is related to improper use of the training phraseology or speaker inconsistency due to stress or other factors may serve as a valuable measure for the instructor to use to identify problem areas in student performance.

— SUITABILITY ISSUES —

Given a basic understanding of speech recognition, the issues regarding the suitability of speech recognition for a specific training application can be addressed. Questions and concerns analyzed at this point should determine whether speech recognition can support the training objectives.

Vocal Characteristics

How important are voiced commands during training? In the kind of training systems we develop, speech recognition is used to simulate human interaction. If voiced commands do not play an important role during real world operations, some other type of input device may be more suitable.

If voiced commands are necessary in the training environment, the next question regards how these commands are spoken during normal operations.

Consistency.

Current speech recognition technology requires vocal consistency. During speaker-dependent training, the recognition device establishes how each person speaks the vocabulary. Drastic deviations from this pronunciation will produce poor recognition. In the Shore Based Radar ATC Training Systems (SATS), trainees may change their pronunciation of unfamiliar words during the course of their training. For example, the word "par" has been pronounced as one syllable when beginning training. As students become more familiar with the phraseology, they change their pronunciation to "p-a-r", spoken as single letters. In order to use this new pronunciation with consistent results, the trainee must re-establish a baseline pronunciation for "p-a-r" in an offline mode before continuing with the realtime training exercise. When developing a curriculum for trainees, the user should be aware that the trainee must be taught the phraseology prior to using it to permit the successful incorporation of speech recognition.

Speech recognition is very robust in terms of speech rate. Most systems can support speech rates from half to double the rate at which the speaker trained. A speaker can speak rapidly or slowly with very good recognition, provided that consistency within the command is maintained. Wide ranges in speech rate violate the requirement for consistency. The phrase "how do you hear me" can be spoken as isolated words "how-do-you-hear-me" or as a rapid stream of syllables "ha-da-ya-hear-me". As long as the speech rate is consistent for a given speaker within the defined constraints, recognition performance will be reliable.

Is vocal consistency required of a successful trainee? With a knowledgeable instructor to monitor and correct poor speech habits, speech recognition will reinforce consistency in speech, since erratic speech will result in poor speech recognition, thereby producing inconsistent system responses. For training applications requiring consistent speech patterns of trainees, limitations of the speech technology actually promote good training. For training applications where vocal consistency is not required and is not feasible in a real world setting, speech recognition in its current state would not be a suitable input device.

Pausing.

Pausing is a natural aspect of human speech. Humans pause to separate ideas within an utterance and to think about ensuing phrases. In a training environment, this characteristic of human speech is more prevalent than in conversational speech, since trainees are in the process of learning while they are speaking. Therefore, any speech recognizer integrated into a training system must perform well with regard to speaker pausing. Some recognizers use pauses to determine the end of utterances. These types of recognizers are not optimal for trainers, since trainees may be cut off in the middle of a spoken command.

Pausing is different from verbal place holding. A trainee may hesitate before a heading by drawing out the word "heading". "fly headinggggggg .....123 . This is not recognizable. A true pause is silence after a word or word group, it is not the extension of the last word spoken. Speech recognizers can accommodate true pauses, they cannot tolerate verbal place holding. In environments where verbal place holding is counter to good operational performance (such as in air traffic control), this limitation is a valuable training tool.

Tone and Prosodics.

Tone is an important element of speech in determining the recognizability of spoken utterances. Monotone speech is reliably recognized by current speech technology, but is certainly not a requirement for good recognition. Natural inflections are supported by speech recognition, and actually enhance consistent performance: a person who speaks naturally is normally very consistent in speech patterns and pronunciations. However, extreme variations in tone due to stress or excitement are not reliably recognized. Users with requirements for stressful training applications where vocal utterances can have extreme variations in tone should consider this constraint when determining the applicability of speech recognition. Naturally if vocal control during stress is one of the training objectives, this constraint can work in the instructor's favor. Note that changes in speaker tone due to nasal congestion related to allergies or colds is well accommodated by available recognizers. Speech devices are surprisingly robust in recognizing speech which, to human perception, has been drastically altered due to nasal impairment.

Training Environment

The training environment is an important area to investigate when determining whether speech recognition is appropriate within a trainer: Is training performed in an environment which can support speech recognition? Topics to evaluate include the interface between the trainee and the training system, the amount of noise in the training area, and realtime considerations.

Trainee Interface.

How the trainee communicates with the training system through vocal commands is of prime importance when determining whether speech recognition can support the training objectives. In order to incorporate speech recognition into a trainer, an interface must exist which reliably transmits voiced data to the speech device. The most common interface is a microphone. For training applications which normally include a microphone such as air traffic control or pilot training, the mobility of the trainee needs to be considered. If the trainee is required to move around the training area freely, some type of remote microphone may be required. Speech devices which allow a remote microphone interface are limited, which may effect the type of speech recognizer needed.

Background Noise.

The noise level in the training environment can impact the decision to incorporate speech recognition in the trainer. For training systems subject to varying levels of background noise, speech recognition may be more difficult than if the environment was subject to constant, low levels of noise. Since microphones transmit all sound, spoken commands as well as background noise may be present. Noise-cancelling microphones are designed to minimize background noise, so microphone selection is crucial for noisy environments. Noise-filled areas can also add to speaker stress during training operations. High stress levels may impact trainee vocal consistency, which impacts recognition. Speech recognizers differ in terms of their capability to handle varying levels of background noise and speaker stress. When determining whether speech recognition is appropriate for a training environment with these conditions, actual demonstrations are a good idea. The various recognizers being considered should be tested in the real world setting to find out whether they can support the training objectives. On the ASATS/SATS system, a recording of the noise generated during ATC operations on a carrier was used to test the robustness of the ITT voice board. In the TOTS tower trainer, it was found that excessive movement within the trainer cabs negatively affected recognition accuracy. This was due to variations in background noise between the student positions, which were open to the visual screen and the air conditioning units, and the instructor area, which was more protected. Installing glass between the student positions and the visual screen improved this situation.

Realtime Considerations.

Realtime training operations require rapid translation of spoken commands to system responses. Long delays between the trainee input and the system response can greatly impact training effectiveness. Speech recognizers differ in their capability to process recognition results in realtime. Recognizers which do operate in realtime provide results continuously with insignificant delays. Non-realtime recognizers generally buffer results for some amount of time, thus delaying the trainee-perceived system response. Although realtime training applications do not eliminate the feasibility of speech recognition, they do limit the types of recognizers under consideration.

Training Phraseology

Applicability.

Another area of analysis is the training phraseology. The training phraseology includes all possible spoken words and word combinations allowable in the training environment. Current speech technology only allows a finite number of words and phrases for a particular application. It is important, therefore, for the user to determine: Does the real world environment for which this trainer is being developed utilize a standardized, identifiable phraseology? If a standardized phraseology does not exist or cannot be developed without decreasing training effectiveness, then speech recognition cannot support training. For training systems which do incorporate a specific phraseology, the phraseology should be identified to further analyze the suitability of speech recognition. This is a critical communication point between the user and the developer. The right combination of commands will achieve high recognition and support training. The wrong combination may support training but poor recognition can make the system unusable. The sooner these tradeoffs are identified and discussed, the better and more usable the trainer will be.

Identification.

The user at this point must identify all permissible voiced commands in the training environment. This identification should include:

- rules regarding how and when each phrase is spoken. For example, all phrases which can only occur by themselves should be identified as such.
- system responses for all phrases.
- special case words. For example, the word "correction" may be spoken within each phrase in order to correct trainee mistakes. Rules regarding these types of words must identified.

Recognition requirements should also be identified at this point. What recognition accuracy is required of the speech device? As has been demonstrated, poor recognition for a particular speaker is not necessarily a bad thing, since it can flag problems in student performance. However, the accuracy for the system should be high when the student is speaking properly. Frequently the user assumes that a high recognition accuracy percentage is the sole discriminator for system performance. In fact, how a system handles a misrecognized phrase is equally important. Misrecognition of certain phrases may be unimportant if these phrases have no system response, misrecognition of other phrases may seriously decrease the effectiveness of the trainer. Recognition accuracy is really only relevant in terms of system performance. The issue of how to quantify system performance is significant, but cannot be addressed here.

Analysis.

The identified phraseology must then be analyzed to determine the feasibility of incorporating speech recognition into the trainer: Can this phraseology be reliably implemented given the speech devices available and the required recognition accuracy? This is really a question for the developer to explore, but the user is also involved, since the outcome will determine whether speech recognition can be implemented. The developer should analyze such things as.

- phraseology size. Given the specified phraseology, the developer must determine whether it will fall within the size limitations of available speech recognizers.
- recognition analysis· The phraseology should be reviewed in terms of optimizing recognition performance. Areas looked at include word competitions, poorly recognized words, and potential word combinations to improve recognition.

The results of this analysis may prove that further phraseology refinement is required. The user must decide whether certain phrases can be modified or eliminated without impacting the trainability of the system. Collaboration between the user and developer during phraseology refinement is crucial in maintaining the effectiveness of the trainer while overcoming any limitations of the speech technology. For training systems which utilize phraseologies that are not alterable and exceed the capabilities of the available devices, speech recognition is not a viable training technology.

— COSTS AND BENEFITS —

When evaluating the use of any technology within a training system, the cost of that technology must be considered. In addition, the user should be aware of benefits inherent in using speech recognition.

Outlays

Speech recognition devices range from $100 to $10,000 in cost. Lower-end devices have a very limited vocabulary capability (100 words maximum), and are generally not robust in terms of handling wide variations in speaker volume and rate. Higher-end models can handle large vocabularies (some over 5000 word limits), and perform well across a wide range of speakers. The requirements of the training application determine the type of speech recognizer needed. Selecting a device which cannot support the training objectives in order to save on device expenses will result in greater expenses during the life of the trainer, since the trainer will not be able to reliably instruct trainees.

The unique requirements of speech technology generate development expenses in addition to the general development costs of the trainer. The training phraseology must be implemented in a way that the recognizer can understand. This may range from 10 labor hours to several thousand depending on the requirements of the speech device utilized. The phraseology must also be optimized in terms of recognition performance across a wide range of speakers.

Speech recognition can incur costs over the life of the trainer. As with any hardware device, the recognizer may need occasional maintenance to replace electronic components. Changes in phraseology over time can also add to lifecycle costs, depending on the time required to translate the phraseology changes into the recognizer's format.

Benefits

If the initial costs of speech recognition are so high, why should it ever be incorporated in a trainer? There are several reasons:

1. Syntax-based speech recognition requires a fixed phraseology and consistent delivery. When a firm knowledge and use of phraseology is critical, this reinforces an essential need.

2. The direct interface between the student and the computer ensures that the system response is precisely in accordance with what was recognized with no assumptions to cause negative training. The computer can also determine the validity of a command with greater accuracy than most humans.

3. Under computer control, a more complex training scenario can be presented without the requirement for significant additional personnel resources to provide necessary interactions.

The result is better and more structured training, using fewer resources.

— CONCLUSION —

Speech recognition has delivered on the promise it demonstrated in the 1970s. Understanding its capabilities and limitations is vital to the successful incorporation of speech recognition into training applications. By evaluating the importance of voiced commands, the training phraseology, and the training environment during the initial requirements phase, the user can make an educated decision on using speech technology within a training application.

— *ABOUT THE AUTHORS* —

Robert Rejent is a computer scientist at Logicon in the speech technology area. For the past 4 years, he has integrated speech recognition into training simulators for the Advanced Shipboard ATC Training Systems (ASATS), the Shore Based Radar ATC Training Systems (SATS), and the Tower Operator Training System (TOTS). He holds a master's degree in Industrial and Systems Engineering from the Ohio State University.

Catherine Meyn manages the Speech Technology Section at Logicon. Since 1978, she has worked in the areas of speech recognition/generation, instructor operator stations and performance measurement systems for air traffic control and flight training systems.

WATERFRONT TRAINERS: LESSONS LEARNED FROM AN EXPERIMENT IN REMOTE TRAINING DELIVERY

Mary L. Sheppe
Naval Training Systems Center
12350 Research Parkway
Orlando, FL 32826-3224

William A. Hayes
Naval Education and Training Program Management Support Activity
Pensacola, FL 32509-5000

ABSTRACT

In an effort to lower training costs, the Navy has initiated a policy to reduce shore based training when the training can be conducted as effectively in an operational environment. To help implement this policy, the Chief of Naval Education and Training (CNET) converted two 30' x 8' trailers to mobile waterfront trainers. It was anticipated that moving the trainers to various sites convenient for ships could reduce the cost of training both in terms of time away from the job and in actual dollars spent on travel and per diem. Each trainer has the capability for delivering training via four types of media: computer based instruction (CBI), interactive videodisc (IVD) programs, videotape (VT), or slide/sound instruction. Training programs cover a wide variety of topics including firefighting, damage control, navigation, safety, reading, math, engineering management, and technical skills. This paper addresses lessons learned from the design, implementation, and operation of this program. Elements discussed include the design of the trainer, the role of instructors, the importance of promotion and advertising to potential users, costs associated with the program, and user acceptance of the concept of providing training in the operational environment.

INTRODUCTION

A Waterfront Trainer (WT) is a 30' x 8' trailer that houses computer stations, interactive videodisc stations, and videotape or slide/sound stations for 10 students (Figure 1). It is a vehicle for delivering remote training. The Navy calls it a WT because it is usually parked on a pier. The trainer is not limited to a pier, however. It could be parked any place that has proper utility hookups--places like flightlines, maintenance facilities, at reserve or guard centers, outside barracks or mess facilities, or even simply by office buildings in parking lots. The concept is to take the training to the student rather than to send the student to a classroom or another facility that is away from the job site.

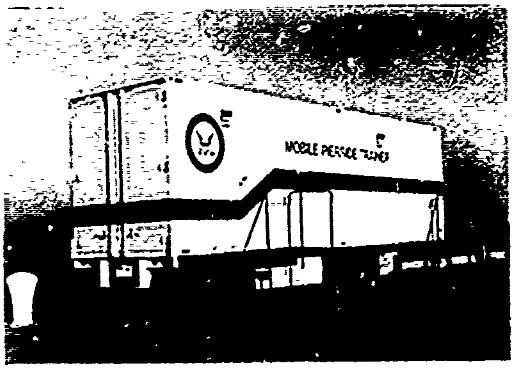


Figure 1. Photograph of Waterfront Trainer (originally called Mobile Pierside Trainer).

BACKGROUND

The driving force behind this program was a continuing reduction in training dollars. In 1985 the Chief of Naval Operations directed the Chief of Naval Education and Training (CNET) to help reduce costs by moving training to the operational environment whenever that training could continue to be conducted effectively. CNET responded by developing two major programs to implement this policy:

(1) Onboard Training (OBT)

(2) Waterfront Trainers (WTs)

The two programs complemented each other. Most of the training programs available for one program were also used in the other program. The purpose of onboard training was to take advantage of the time sailors have on deployment at sea by making a variety of training programs available in a stand-alone mode.

The waterfront trainers were designed to accomplish three purposes. The first was to provide readily available training on the pier at a time convenient to the students. This would allow some of the time in port to be converted into training time without requiring the student to be away from his job for long periods of time and could help save training dollars spent on transporting students to schools or other training sites. Secondly, having the trainers in port with a resident instructor gave shipboard personnel the opportunity to become familiar with the hardware and the training programs so that they would be more likely to utilize the

programs during deployment. Finally, the WTs were to serve as a test bed for newly developed programs prior to the release of those programs to the fleet for OBT.

The WT concept also afforded some other potential benefits. In most job situations (and certainly onboard Navy ships) computers that can be dedicated for training purposes (even on a part time basis) are limited. Having the trainer available in a central area allows many in-port units to use the same training equipment. It allows student access to a computer near the work site, yet far enough away that interruptions are less likely. The WTs have some training programs that are not available onboard ships. Finally, more supervisors may willingly support training because it is easier to send a person to training for a few hours than a few days.

The Navy has two WTs. They were prototypes that were placed into service in early 1989. The trainers were located in Norfolk, VA and San Diego, CA. In December, 1989 the West Coast WT was moved to Long Beach and has since remained there.

Six Navy commands combined their efforts to implement the WT program. In August, 1988 they signed a Memorandum of Understanding outlining the responsibilities of each of these groups. The Commanding Officer, Naval Education and Training Program Management Support Activity (NETPMSA) in Pensacola, FL. became the manager of the pilot program. The Naval Training Systems Center (NAVTRASYSCEN), Orlando, FL was tasked to evaluate the program. The evaluation addressed the concept of remote training at pierside and covered the prototype phase (January-July 1989) and a follow on period of one year (July 1989-June 1990). (Analysis of the effectiveness of the individual training programs was not covered in these evaluations). NAVTRASYSCEN published two technical reports that cover those evaluations in depth. This paper discusses lessons learned from these evaluations and chronologically covers all phases of the WT program from design through user acceptance. Information presented can provide insight useful for others considering this type of training.

DESIGN

Selection of Vehicle for Training Delivery

The first decision regarding design of the WTs was choosing between a fully motorized, self-contained van (similar to a mobile home) and a traditional trailer (requiring a tractor for transport) like the WTs. There were advantages to both types of vehicles. First, trailers cost less to purchase and less to maintain because there is no engine. Also, there is no cab taking up space that could be used for additional student stations.

The primary advantage to a motorized, self-contained van would have been increased mobility. The evaluations showed that relocating the trainers did increase utilization (Figure 2) and relocating would have been easier with a motorized van than it was with a trailer. Moving WTs required coordination with local agencies to obtain a tractor for moving and it was sometimes more difficult to identify a site because the trainer would be parked there for an extended period of time. Lost training time was also a factor. It took more time to break down, secure, move, and set up again with a trailer than would have been required with a fully motorized van. Finally, more frequent moving would have increased the visibility of the trainer which in turn could have increased student interest and utilization.

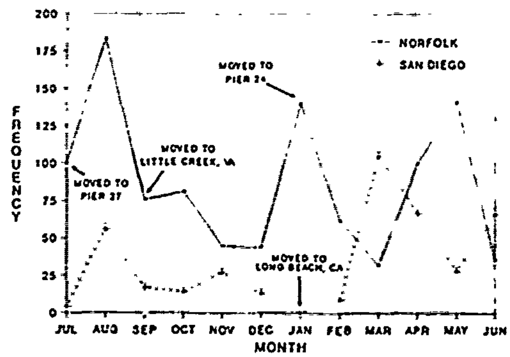


Figure 2. Impact of moving the WTs on student throughput (Phase II Evaluation).

Design of Workstations

Workstation configuration was the next matter to be considered in the design of the trainers. For the WTs, the instructor station was placed in the front of the trainer and the 10 student stations lined one side wall and the rear wall. This design allowed easy access to all stations, provided one wide aisle with easy access to both exit doors and provided students space for working together. The workstations consisted of:

6 Student Stations with Capability for Computer Based Instruction (CBI) Training Programs Only

2 Student Stations with Capability for Both CBI and Interactive Videodisc (IVD) Training Programs

2 Student Stations for Videotape (VT) and/or Slide and Slide/Sound Training Programs

1 Instructor Station

Each CBI station consisted of a Zenith Z-248 computer with a 20-megabyte hard drive, two 360 kilobyte floppy disk drives, 640 kilobytes expanded memory, an EGA card, a 13-inch color monitor and headphones. The two interactive videodisc stations were similarly equipped except for a multi-sync EGA card, a 1602 Visage board, a Pioneer LD-V4200 laserdisc player and a 13-inch Visage touch-screen monitor.

The two videotape carrels each had a 1/2 inch VHS tape player, a 1/2-inch BETA tape-player, a 3/4 inch U-matic tape-player, a 13-inch color monitor and headphones.

The instructor station in each trainer was equipped with hardware identical to that in the student CBI stations and additionally had an internal modem, a telephone, and a printer.

Other Elements of Internal Design

Other items that required consideration during the design phase included:

Lighting: Lighting had to be sufficient for students and instructors to work. Screens to protect against glare were purchased for monitors in both trainers. Radio frequency filters were installed to prevent problems with the fluorescent lighting that could be caused by interference from shipboard radar and sonar radiation patterns.

Noise Control and Interference: It was anticipated that noise would be a problem especially when trainers this compact were operating at full capacity. Effective use of sound dampening materials was therefore important and headphones for VT programs and computer programs with audio were essential.

Security: Security involved not only the physical security of the trainer per se but also security for the software, the documentation, the hardware and the equipment inside the trainer. Secure locks were installed on both doors and the hardware/equipment items were secured with cables. In addition to concerns for security in the design phase, this was a factor that required examination each time a trainer was relocated. The WTs have mostly been on Navy stations where access is controlled. Provisions for security would vary with each site location and could also vary with operating hours. For example, having a trainer open after dark would necessitate provisions for outside security lighting.

Safety: Standard safety items such as two exits, railings on the stairs, smoke alarms and fire extinguishers were applicable to these trainers. Each trainer also had a first aid kit. Interior safety design factors included ensuring that rollers and backs on student and instructor chairs were stable and that equipment was securely installed and not likely to fall and cause injuries. Like security requirements, safety issues must be re-examined each time a trainer is relocated. For example, special provisions applied when a trainer was parked in an area where ordnance was being handled. Weather sometimes caused safety problems. In Norfolk, the trainer had to be closed on a few occasions because of instability in extremely high winds.

Working Space: Having a student sit at a computer did not eliminate the need for an area to write, take notes, etc. The space in the WTs was limited and some additional workspace was needed. The amount required depended on the training program the student was using. For some of the programs, especially those requiring navigation charts, students needed a substantial workspace. Lap boards served as desk space for small jobs and the wall served as a space for larger requirements, e.g., a place to put navigation charts.

Storage Space: CBI programs require documentation, sometimes substantial documentation (manuals, user guides, etc.). For the WTs, shelves were built over each student station and documentation was readily available as required. Shelves and storage cabinets were also required to store the videotapes and slide programs.

Climate Control: Climate control units should be sufficient to handle the load, be stable, have reliable thermostats, and have output/intake vents strategically located. The design of the climate control units caused minor problems for both WTs. Instructors in both WTs regularly opened and closed the doors to help maintain the temperature.

IMPLEMENTATION

Procurement of Vehicles

Once it was decided to use trailers rather than motorized self-contained vehicles, two units were obtained from War Surplus Storage (Construction Battalion Base, Gulfport, MS) at no cost. The trainers were refurbished and reconfigured by the Naval Public Works Center in Pensacola at a cost of approximately $85,9000.

Site Selection

Norfolk, VA and San Diego, CA were selected as sites for the prototype trainers. These sites are major ports with potentially large student populations, and four of the six Navy commands involved with the implementation of the WT program have their headquarters at these locations.

Instructor Qualifications

Instructors were contracted from Central Texas College. The term "instructor" actually meant "facilitator" in this case. Instructors were not subject matter experts (SMEs) for all the training

programs available in the WTs, nor could they be expected to fulfill that role. Nevertheless, knowledge of computers and the ability to assist novice operators in the use of the equipment and software were essential qualifications. Being able to work with many different types of people and having the ability to troubleshoot problems associated with computer training were also necessary skills. Additionally, instructors were required to have some knowledge of the Navy and the skills required on Navy jobs. Instructors were trained on the hardware in the WTs and given an opportunity to become thoroughly familiar with the training programs prior to working with students.

Advertising and Promotion

Advertising and promotion were important elements in the implementation phase and they continued to be important elements throughout the operation of the program. One lesson the Navy learned was that advertising must be an ongoing program. It was especially important in the WT program because the student population was constantly changing -- for example, ships routinely coming and going. Training at the WTs was optional and this increased the need for advertising and promotion to continually attract students. Advertising and promotion activities were handled by the local agencies on both coasts. Promotional efforts included: articles in base and local newspapers, announcements at meetings, briefings, special orientation visits to the trainers, distribution of the Waterfront Directory of Training to all ships, messages to potential users, flyers posted in libraries, and distribution of brochures.

Personnel managing the WTs on both coasts thought that the most effective advertising methods were the quarterly messsages sent to the fleet and the promotion given the trainer at meetings (especially meetings involving senior officers). More students reported hearing about the WT from their supervisor than by any other means. A substantial number of students also reported that the Plan of the Day, instructions, and other students had been key sources of information regarding the WTs. Thus, while news media and brochures may have a place in promoting these trainers, advertising through the chain of command was the most effective method of reaching potential students.

Training Programs for the WTs

There were several interesting lessons for the Navy about the selection and use of training materials. First, the selection of training materials to use in remote trainers was a crucial element. A perfect facility will not offset poor or boring training programs. Second, as anticipated in the planning phase, the number and types of programs made a real difference. When the trainers opened there were 59 programs available. Programs were added continuously and as the number of programs increased so did utilization. By the end of the second evaluation period there were 98 programs available at the trainers.

Utilization by Media

CBI was the most popular medium with students during the two evaluation periods. Table 1 provides information on the number of training programs in the media categories and Table 2 provides the utilization by media for both phases of the evaluation. It should be noted that the number of interactive videodisc programs was limited, especially when the program was initiated. Therefore the importance of IVD programs in this type of training environment deserves further analyses.

Table 1
Number of Training Programs
by Media Categories

Media	End of Phase I (Jul 89)	End of Phase II (Jun 91)
Computer Based Instruction	30	43
Videotape	26	29
Interactive Videodisc	5	20
Slide or Slide/Sound	6	6
TOTAL	67	98

Table 2
Utilization by Type of Media

Media	Phase I (Jan-Jul 1989) (percent)	Phase II (Jul 89-Jul 90) (percent)
Computer Based Instruction	68	55
Videotape	20	30
Interactive Videodisc	8	9
Slide or slide/sound	1	1
Not Reported	2	4

Utilization by Subject Categories

Training programs selected for the WTs covered a wide range of topics including management and administration, firefighting and damage control, navigation, communication and recognition skills, equipment operation, basic skills (reading, math) and self-improvement. As noted, increasing the number and types of programs available attracted more students. In the prototype phase of the WT program, students represented only 22 out of more than 80 jobs (ratings) in the Navy. By the end of the second evaluation period with an additional 39 programs, the number of ratings represented had more than doubled to 50.

Some programs were used consistently throughout both phases of this evaluation. These programs included Maneuvering Board Refresher Training, Flags, and Rules of the Nautical Road, all of which deal with skills required by a large number of Navy personnel. Sometimes a program seemed to "catch on" in one location and be totally ignored in another. For example, the videotape entitled "Beating the Odds -- Samuel B. Roberts: Fight for Life" accounted for 8.2 percent of the total utilization on the East Coast during the Phase II evaluation, but was not used at all on the West Coast. Likewise, 5 of the top 10 programs utilized on the West Coast did not make the top 10 on the East Coast.

Firefighting and damage control programs accounted for approximately 25 percent of the total utilization of training programs in both phases of the evaluation. This was probably because these are important skills for every sailor. In Phase II, 18 programs on equipment operation were added and utilization in this category increased immediately. Table 3 provides information on the utilization of training programs by subject categories for both phases of the WT evaluation.

Overall, there was some interest in a variety of subject categories (including, for example, programs on SAT Preparation, Weight Loss, and Physical Fitness). However, the most used training programs were those that could apply directly to the student's job or those that taught a critical skill (such as firefighting).

DAILY OPERATIONS

Daily Procedures

The day-to-day operation of the trainers was relatively trouble free. The instructors were responsible for:

- -- scheduling students
- -- accommodating and assisting walk-ins when possible
- -- supervising students
- -- cleaning the trainers
- -- taking inventory and maintaining the security of the of the software and hardware in the trainers
- -- reporting problems to appropriate management personnel.

Recordkeeping was limited to recording student demographic information, the time in/time out, and the training programs utilized.

Changes in Operating Procedures

Once the trainers were operational, there were several observations or events that precipitated changes in procedures for conducting business. For example, operating hours were reduced substantially. Originally the trainers were open for 14-16 hours a day. The plan had been to accommodate students both during duty and

Table 3
Utilization by Subject Categories
Phases I and II
(Percent)

Subject Category	Number Programs Available (End of Phase II)	Utilization Phase I	Utilization Phase II
Firefighting/Damage Control	15	24.3	28.9
Communication/Recognition	11	17.0	21.1
Navigation	5	29.8	19.0
Basic Skills/Self-Improvement	26	22.9	14.5
Management/Administration/GMT	9	2.8	5.6
Equipment Operation	24*	.2	3.6
Safety	8	.6	3.0
Not Reported By Students		2.3	4.3

*18 of these 24 programs were not added until October 1990.

off duty hours. Student arrival times were recorded and it was determined in the first phase of the program that the trainers were not being used sufficiently during off duty hours to justify the costs. Table 4 shows student arrival times for the period Feb-Jul 1989 for both coasts. Based on these data, hours were reduced as soon as the contract for the instructors was renewable, leading to significant cost savings in salary dollars.

Table 4
Student Arrival Times
(Feb-Jul 1989)

	East Coast	West Coast
0700 to 0859	137	23
0900 to 1059	86	32
1100 to 1259	82	44
1300 to 1459	73	55
1500 to 1659	14	5
1700 to Closing	53*	3
Not Reported	11	0

*38 were in February 1989 and concentrated in a one week period with nearly all personnel being from the same ship.

One other significant change during the operational phase was the decision to use military personnel as instructors in Long Beach rather than contracting for civilian instructors. There were both advantages and disadvantages to having military instructors. First, it was more economical in this case (although each situation could be different depending on the rank and salary of the military person). Paying civilian instructors was the largest single line item in the budget for operating the WTs. On the other hand, civilian contract instructors did not have other duties and were able to be on site fulltime. Students and management personnel were complimentary about both military and civilian instructors. It was concluded at the end of the evaluations of the WT program that the decision to use military or civilian contract instructors should be made by local personnel responsible for the management of each trainer. Factors that should be considered in making this decision would include the operating hours, the need to have a full time instructor, the costs involved, and the anticipated student load for that area.

Amenities

Two additional factors that required consideration in the operation of the WTs were the need for break areas and access to parking in some cases. Trainers of this size obviously cannot come equipped with smoking areas, vending machines, and bathrooms -- amenities that can have a positive impact on a training program. When the trainer could not be parked within a reasonable walking distance of suitable facilities, the host organization provided a portable restroom, and a roving snack bar provided food and beverages. Parking was a consideration if the students were not within walking distance. For the WTs, some students were transported by Navy vehicles, some were within walking distance, and some drove their own vehicles.

SUPPORT REQUIREMENTS

Personnel on both coasts described the trainers as very low maintenance facilities. As previously noted, the trainers were not equipped with engines which would have increased the need for maintenance. Also, the absence of water and restrooms further decreased the requirement for maintenance. The host activities provided telephone and utility hookups and the WTs experienced no substantial problems in these areas.

The need for supplies was minimal. The trainers operated with one printer requiring a limited amount of paper, one telephone, a few pencils, a log book, and various cleaning supplies. The hardware and software in the trainers were also low maintenance items during the evaluation periods. However, it was noted in the evaluations that these items were relatively new and that maintenance and replacement costs would have to be considered in a long range plan for remote trainers.

CONTROLLING COSTS

The one question that is usually asked about any training program is: "How much does it cost?" Some costs are inherent and cannot be reduced substantially but in some areas there are ways to control the costs. The start up costs for the two WTs (nonrecurring costs such as hardware, software, trailers, and transportation) were $248,000 (total costs for both trainers). These costs included approximately $123,000 for training programs (CBI, IVD, and videotapes).

Following the prototype phase, training programs were added continuously. Some were developed by the Navy or through a Navy contract. Several were purchased off-the-shelf from commercial vendors. Others were obtained from military sources at no charge. Salaries for instructors remained the largest single recurring cost. As discussed, the West Coast was able to reduce this cost by using military personnel. While the individual's salary was still an expense, it was less than hiring civilian contractor instructors in this case. Annual recurring support costs (instructors, utilities, and maintenance) for the trainers during Phase II were approximately $47,000 per trainer. This figure was expected to decrease further on the West Coast because of the changeover in instructors.

Increasing Efficiency

Because the initial utilization rates were lower than expected the cost per hour of training was substantially higher than what was desirable. Achieving efficiency in the WT program meant increasing the utilization and thereby decreasing the cost per student or cost per hour of training. Both evaluations of the WT program identified several obstacles to overcome in order to increase utilization. These included:

(1) Ships in port have heavy workloads and busy schedules. Time to train for shipboard personnel is severely limited because getting ready to return to sea has a high priority. There is probably little the WTs can do about this. Therefore, it became even more important to ensure that the training offered in the WTs would be of relatively high value to the user.

(2) Training in the WTs did not specifically fill mandatory training requirements and therefore did not receive a high priority vis a vis operational requirements. Training was voluntary and "nice to have." Only in a very few cases could a supervisor send a student to the trainer and subsequently "check off" a requirement. Therefore, a project was initiated to match training programs to training requirements.

(3) The trainers needed to be relocated more frequently in order to accommodate different ships and more students. Although the trailers were not as readily mobile as motorized self-contained vehicles, relocating was quite feasible, both from a logistic support standpoint and a cost standpoint.

(4) Too many people did not realize the training was available. There was a need for innovative and continuing advertising programs to reach the changing student population.

(5) Continuing to increase the inventory of programs available in the WTs was important. New and additional programs offered increased training opportunities for returning students and were also helpful in attracting new users.

USER ACCEPTANCE

User acceptance is an important measure of the success of any training program. Studies and evaluations can say that training is effective, that it teaches what it is suppose to teach, and that students who complete the training can do the job. Nonetheless, a training program will not succeed if the end user does not perceive that there is value in the training. The user has to see enough value to justify letting people off the job and sometimes see enough value to pay some of the costs.

For the WTs the acceptance level appeared to be somewhat of a mix. Looking at actual utilization rates, the picture was dismal. However, looking at what the users said about the trainers presented a different impression. In other words, utilization was low, but those who trained at the WTs and personnel involved in the operation and management of the program thought there was substantial value in continuing and expanding the program.

Management personnel generally thought that the WTs were providing a valuable service to the fleet. Instructors reported that students were serious, enthusiastic, and learning from the training programs. Students and management personnel reported that the instructors were knowledgeable, cooperative, and helpful. Students reported that the content of the training programs was at the appropriate level. They said their level of understanding of the subject increased and that they expected this training to improve their performance on the job; Figures 3 and 4 reflect student responses to these two areas for Phase II of the evaluation. In both phases of the evaluation, students rated climate control, noise level, the amount of workspace, cleanliness, and hours of operation as adequate, good or acceptable in nearly all cases. Student comments on the log sheets and on evaluation questionnaires were overwhelmingly favorable. Over 89 percent of the students surveyed said they would be willing to return to the WTs for additional training should time and schedule permit.

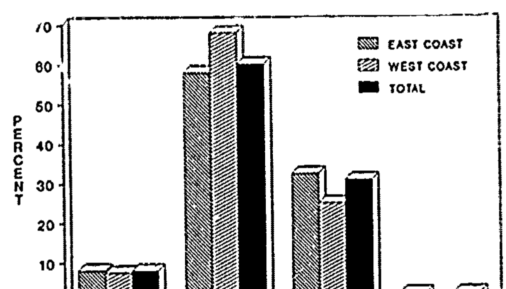


Figure 3. Student perceptions of the extent their understanding of the subject was improved by training at the WT (Phase II Evaluation).

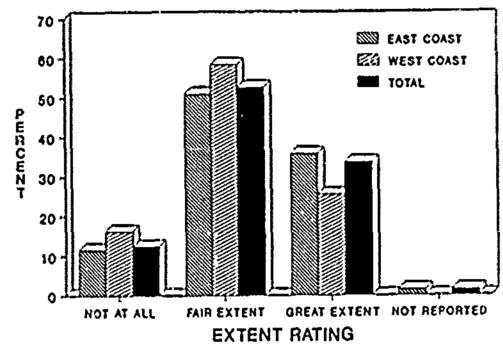


Figure 4. Student perceptions of the extent their training at the WT may improve job performance (Phase II Evaluation).

SUMMARY AND FUTURE DIRECTIONS

While the WT program did not initially fulfill its potential in terms of student throughput, it did fulfill a training need and has continued to do so. The WT prototypes were an experiment. They were not intended to solve all training problems, but to test the concept of training in the operational environment.

The trainers are providing training that is not readily available from other sources and students are benefiting from the experience. The trainers are also successfully serving as a testing ground for some training programs prior to distribution to the fleet.

Currently, the WTs are still located in Norfolk, VA and Long Beach, CA. There is an ongoing effort to add training programs as they become available. The total number has grown from 59 to more than 120. New technologies are providing the potential for converting additional training packages to video, CBI, IVD, and for storage on CD-ROM. Utilization of the trainers tends to rise and fall with the concentration of ships in port and overall has been showing a gradual increase. Advertising and promotion efforts continue and the Navy is aggressively pursuing ways to increase utilization and obtain maximum benefit from this program.

REFERENCES

Department of the Navy. (1988, August). Memorandum of undestanding: Prototype mobile pierside trainers (MPTS). Pensacola, FL: Naval Education and Training Program Management Support Activity.

Department of Navy Contract (1988, December). Contract Number: N00612-88-0488. American Technical Institute of the American Educational Complex. Charleston, SC: Regional Contracting Department, Naval Supply Center.

Department of the Navy Contract. (1990, January). Contract Number: N00612-89-C-0169. American Technical Institute of the American Educational Complex. Charleston, SC: Regional Contracting Department, Naval Supply Center.

Sheppe, M.L., Sheppard, D.J., Chuderwicz, R.P., & Dunlap, S.W. (1990, January). Mobile pierside trainer - - Evaluation of prototype phase (Technical Report 90-002). Orlando, FL: Naval Training Systems Center.

Sheppe, M.L., Sheppard, D.J., & McDonald, R.G. (1990, October). Waterfront Trainer Program - Evaluation of Phase II (Technical Report 90-017). Orlando, FL: Naval Training Systems Center.

Mary Sheppe is an Education Specialist in Instructional Systems Group (Surface) at the Naval Training Systems Center, Orlando, FL. She holds a B.S. degree in History and Education from Radford University in Virginia and a M.S. degree in Instructional Systems from the University of Central Florida. She has been at the Naval Training Systems Center since 1984 and was head of the project team responsible for evaluating the waterfront trainers.

William Hayes is an Education Specialist in the Instructional Technology Implementation Office at Naval Education and Training Program Management Support Activity, Pensacola, FL. He holds a B.S. degree in Mathematics from Florida State University, and a M.Ed. degree in Educational Leadership from the University of West Florida. He has been with the Naval Education and Training Command since 1966 and was program manager in charge of the development and implementation of the waterfront trainers.

DOES THE FLIGHT SIMULATOR USER KNOW WHAT HE HAS GOT?

WING COMMANDER R M PROTHERO MRAeS MBIM RAF

ROYAL AIR FORCE

ABSTRACT

The object of this paper is to make Users, Managers and the Synthetic Training Industry, think again about the utility and employment of modern flight simulators. Does the simulator User really know what he has got? Flight simulators have a long pedigree as training devices, and they make a very significant contribution to safe and cost effective training. As a consequence, it is not surprising that they are firmly labelled as "training" devices. However, as technology has given the User ever more capable machines, it is suggested that simulators have outgrown the "training" image. Modern advanced simulators are now so capable that they rank on a par with their parent flight machines. Regarding simulators in this light opens up a new concept for their use as flight experience developing machines. This would be of particular benefit to the military, but even the civil community could accelerate crew experience gathering if the will to do so were present. Such a change in the concept of employment of advanced flight simulators would open up further avenues for their use, and could affect the design specification for future advanced simulators. The growing acceptance of simulators as an essential and integral part of initial flying training, and the concept of Mission Rehearsal, indicates that the User even now almost accepts that simulators provide experience as well as training. Now the User should look again at the basic concept of employment of modern advanced flight simulators to ensure that the true capability of the machines is fully exploited. This Paper represents the views of the Author, and does not necessarily represent the views of the UK Ministry of Defence.

INTRODUCTION

Modern flight simulators have their origins in the Link Trainer, and basic training simulators well documented over the years. All were designed specifically as training equipment. In general, the User's view of simulators has progressed based on its historic employment in training organisations which, in turn, were structured on what had gone before. Flight simulators have been firmly accepted as "training" machines alone. However, following the rapid technical advances in simulation, it is opportune to make a fundamental reappraisal of the advanced simulators in-service to see if the restrictive "training" label could be removed, allowing them to be considered as experience gathering machines on a par with their host aircraft. It is considered that such a new concept of employment could lead to very significant benefits to the User community, both civil and military. The nature of modern advanced simulators, their use as experience gathering devices, and the consequent advantages will be discussed in this paper.

THE NATURE OF MODERN SIMULATORS

Firstly, let us look in more detail at the nature of modern simulators. The universal demand for more cost-effective aircrew training in recent years has lead to the User constantly requiring more capable high fidelity simulators. The synthetic training industry has responded to the demand with better fidelity provided by faster computing, more accurate aircraft aerodynamic modelling, digital control loading, improved motion systems and so on. In addition, visual systems have been developed more accurately representing the outside world, and the simulation of external environmental effects, from micro-bursts to fog has become possible. To meet training requirements, the User has continually sought greater technical innovation, and has then used enhancements to allow ever more comprehensive training at a relatively reducing cost. However, User, Managers, and Industry, have not generally appreciated that the very nature of the flight simulators they have created has changed in the process of adding more capability. They are no longer simulations of aircraft capable only of "training", they are effectively front-line operational equipment as challenging as an aircraft, and this applies particularly to the military simulator. In the civil environment, training needs are somewhat different, and the simulators reflect this, but the basic tenet is still true. Modern simulators are just as much a flying machine as the aircraft itself, and in many cases, they are more capable as they are not restricted by airframe fatigue, peacetime flight limitations, or flight safety considerations. Moreover, they can now be operated in any environment at will.

There can be little doubt that modern simulators are "flying machines" to all intents and purposes. The Harrier GR5 full mission simulator for example, described in detail in a paper presented at the 12th IITSC, comprises a replica aircraft cockpit with motion cuing in a 24 ft dome. It has a unique advanced visual ESPRIT display system, and is mounted on a six degree of freedom hydraulic motion platform. The visual scene is provided by a Modular Digital Image Generator (MODIG) system capable of generating several hundred realistic ground targets of many types. One channel provides a wide angle, low resolution peripheral background scene; a second channel provides an eye-slaved, high resolution area of interest foveal scene,

whilst a third channel produces imagery for the angle-rate bombing system head-down display. A fourth channel produces imagery for the forward looking infra-red head-up display. The visual system data-bases for the simulators already in existence cover a large part of the Southern United Kingdom and the Central German region. Airfields, major landmarks, weapon ranges and key points are all modelled in significant detail so that they are instantly recognisable, as indeed are all the targets, both fixed and mobile. Moreover, the visual system databases may be updated or enhanced off-line by military personnel using a separate workstation.

Turning to its synthetic operating environment, the simulator has a complex suite of sub-systems to provide a realistic and complex electronic warfare scenario in which missions may be flown, and real world threat data may be incorporated within the system and fully integrated, thus allowing full mission rehearsal as defined at IITSC in 1989 by R Wiggers Et Al. The system is also fully NVG compatible, and NBC protection equipment can be worn throughout sorties. The simulators will in practice be the only safe environment in which pilots will be able to fly with such equipment. But they do have one limitation at present, they are individual simulators, and thus cannot allow formation sorties to be undertaken. However, networking the simulators is a possibility. Alternatively, it may be possible to provide computer generated "intelligent wing-men" in the future to allow formation tactics to be employed. Incidentally, the off-board Instructor Operating Station comprises an advanced system incorporating full replay and monitoring. In addition, a Remote Debrief Facility is available and can replay sorties up to one and a half hours duration off-line. This is a description of just one advanced simulator, but there are others in existence, or build.

Looking at such simulators, and the synthetic environment in which they operate, it is clear that the machine is in every way potentially as capable as the aircraft, and in peacetime can provide a fully representative mission environment. This is not just a "training" machine which is capable of specific mission rehearsal, it is a machine in which operational aircrew can fly sorties which are just as challenging as a real flight. Thus, the development of semantic and motor memories of task related data is possible, together with the acquisition of situational awareness skills. This is only one example, but simulators now at the specification stage, for example for EFA in Europe, the F 22 and so on in the USA, will have even better features for replicating the flying environment, and will be even more akin to the aircraft in every respect except in "adrenalin" factor, and the application of sustained "G", but its effects will be replicated using motion platforms, "G" seats, and "G" suits.

Although studies of the "training" effectiveness of simulators, with and without visual systems, have been carried out in the past, it is the RAF's intention to do a full analysis of the full "operational" capabilities of our Harrier simulators when they enter service later this year. To our knowledge, this will be the first time that a simulator has not just been placed within a training system, but has been assessed giving students and experienced pilots, not only the simulator hours specified by the Harrier conversion course, but eventually variations on the simulator to flying ratio to see if the output standards can be improved through the greater use of simulation. Moreover, it will be assessed as an experience gathering machine for squadron pilots in the low-level attack role, as will the new German Air Force Tornado simulator specifically configured for low level training.

THE SIMULATOR AS AN EXPERIENCE GATHERING DEVICE

If advanced simulators are accepted as a means of performance enhancement through the more rapid accumulation of experience that earlier could only be gained in the air, several other benefits emerge. The most important point is that, at present, there is a marked and understandable reluctance to reduce aircrew flying hours, whether the crew are in training or operational. Against this, flying becomes ever more expensive, budgets tighter and much training is environmentally unfriendly. This reluctance could now be outdated. Advanced mission simulators challenge the pilot just as much as flying the aircraft, and there is no reason why the crews should not be scheduled to fly the simulator as often as they have time to do so. It is not suggested that flying should be cut back significantly for those who need it. However, over the years it has been accepted that more experienced aircrew need fewer flying hours to remain proficient than their younger colleagues. Thus, by bringing forward the experience proficiency curve, the financial and flight safety benefits of this reduced individual training requirement could be spread over a greater number of flying years.

In conflict with this, it has been suggested that more simulation means less motivation, but this view must be questioned. The motivation of civil and military aircrews depends on many varied factors from financial reward to status, and from excitement to technical achievement. An emotional view is that pilots wish to leave the bonds of Earth. This may have been true in the days of Barnstormers, but modern generations brought up in the age of computers and automation see things slightly differently, and accept computer simulations in a way that the older pilot finds difficult to appreciate. In the RAF for example, our Tucano simulators are readily accepted by our students despite the misgivings of the more senior staff. In other words, the

weight of reward is biased towards technical skill and achievement, which in some cases is as well realised in an advanced simulator, as in the air.

BASIC FLYING SKILL DEVELOPMENT

The retention of basic pilot handling skills is essential for both civilian and military crews. Its vital nature can be seen from recent Boeing 747, Boeing 757, A320 and DC10 incidents, quite apart from some of the casualties of the Gulf war. It has been argued that flying simulators, rather than the aircraft, detracts from basic skills - probably due to the poor fidelity of earlier simulators, but this is no longer true. However, the opportunities to practise basic flying skills in the air are becoming less with the advent of glass cockpit instrumentation, advanced system automation, and fly-by-wire. In addition the pool of experienced civil and military "hands-on" pilots who have the experience to teach such skills is decreasing with each passing generation. Now, in many cases, the modern simulator is the only vehicle in which basic skills can be practised after initial training. Indeed, it is an ideal vehicle in which to retain handling experience as it can be flown safely with automatic systems disengaged, in configurations seldom, if ever, flown, and in atmospheric extremes possibly otherwise only encountered once in a pilot's flying career. In the light of this capability, aircrew simulator tasking should no longer be set in terms of the minimum to remain competent in procedures, emergencies or weapon system handling. Aircrew should spend more time in the simulators and go beyond mere competence training and be taken into the realms of experience acquisition. Civil flying is clearly mainly revenue producing and is not therefore a commodity to be cut to achieve savings, and civil simulators are generally fully utilized. Nevertheless, to elevate the standards of the less experienced civil aircrew of the future, additional simulator capacity would be a good investment for a progressive airline. In contrast, many military simulators do have spare capacity that could be used for experience enhancement if the machines were of the necessary calibre, and the sponsor tasked the crews accordingly.

FLIGHT SAFETY IMPLICATIONS

As stated earlier, other avenues of advantage emerge from a change in the concept of employment of advanced simulators. Let us look first at some flight safety aspects. At present, simulators are used for 2 types of training, initial conversion, and continuation training, usually featuring mainly emergencies. Initial student errors in the simulators are not usually recorded as they are regarded as "training" errors which can be eradicated, rather than operating errors. Consistent errors in continuation training are sometimes fed back into the flight safety loop, but usually in an informal way. In contrast, aircrew errors in aircraft receive a great deal of attention Why this inconsistency? Errors in modern advanced simulators, regardless of the experience level of the crews, are indicative of what will happen sooner or later in the air. Here, CHIRP (or ASRS in USA) reports are pertinent. A recent incident in which a crew failed to identify correctly an engine malfunction in flight, and the aircraft subsequently crashed, is relevant. This error was not uncommon during initial training on the type, but it was thought that once the crew had been "glass cockpit" trained, and could trouble shoot correctly under test conditions, the problem had gone away. If the simulator had not been regarded purely as a training device, but just as much a piece of flying equipment as the aircraft, the consistency with which crews had difficulty translating the information presented to them would surely have alerted supervisors and management to the fundamental problem, and that an accident was probably just waiting to happen.

Another benefit would arise if advanced simulators could replicate flight profiles from accident data recorders. It would then be possible to recreate accidents under the operational circumstances under which they actually occurred. This would allow the current imponderables of ergonomic factors, human factors, distraction, and so on, to be analysed in much greater detail than is possible at present, and their relevance to the cause of the accident more readily assessed. For a crew sitting in a simulator being driven by accident tapes, this would be the ultimate in learning by others experiences!

COCKPIT RESOURCE MANAGEMENT TRAINING

Another area where a change in the concept of flight simulator usage is relevant is in the area of cockpit resource management, or crew co-operation training. It is a topic that has attracted much attention since several accidents have been caused by lack of crew co-operation. The topic is vitally practical, but most training courses depend more on classroom work than practical lessons. In fact, it has been suggested to the Author on more than one occasion that it would be demeaning for experienced aircrew to be put into a "training" simulator to be taught cockpit resource management. With a modern simulator, nothing could be further from the truth. The place to learn cockpit resource management, develop strengths, and eliminate weaknesses, is in a cockpit with active participants, and in scenarios that are easily generated in our modern machines. In the past, good aircraft and crew captains were generally the older and more experienced. This need no longer be true. Now, if the simulator were regarded as a post-graduate tool, and not just a trainer, relevant experience could be acquired through greater use of simulators. This of course implies that aircrew will operate the simulators in the same way as the aircraft, and this is not true in many cases at present. But need this be so?

FUTURE SIMULATOR DESIGN REQUIREMENTS

If it is accepted that simulators can function above the level of "training", could their performance as experience gather machines be improved by specifying additional features, without losing their training value? Basically there can be no reason why this should not be possible. Simulator sortie scenario generation could easily be made transparent. The crew need only know the task and the weather, as few actual flights start with a formal instructional brief. This would then help to engender in crews a similar mental attitude in a simulator, to that in the air, and behave accordingly. It is of interest here that at least one major simulator manufacturer has gone to great lengths in their latest range of simulators to make the machines appear more like the aircraft, both externally, and in the total flight deck area, to improve the ambiance and realism of simulator sorties. Nevertheless, as stated earlier, based on many years of the Author's instructional experience, there can be little doubt that crews do not normally behave in the same way in the simulator, as in the aircraft. They know they are under scrutiny of an operator or supervisor, and go to great pains to conform to standard operating procedures, or display their self-perceived skills. As with aircraft, future simulators should be capable of being operated by the crews alone when required. They would then be more likely to adopt their normal operating patterns, check list usage, target speeds, look out patterns, and so on, that they would normally employ in the cockpit.

There is a very high degree of system and performance fidelity in our simulators, but it is suggested that simulators could be enhanced even further if the machines were developed to produce all the random inputs and uncertainties that occur in the air without the immediate presence of an instructor. For example, weather effects could be on a random basis within an overall scenario, conflicting traffic generated at random, and so on. Equally, the existing capability of the simulator to generate system failures or emergencies when cued by the flight profile would become more important. A discrete video recording system would also be of value in self de-briefing crew co-operation, lookout, drills, and settling arguments! Most aircrew learn by their errors, and are quite capable of de-briefing themselves with such a system. The enhancement in the potential of simulators with such minor modifications would more than recompense any slight additional cost that the User would have to bear.

FUTURE SIMULATOR PROCUREMENT

This then touches on simulator procurement and the development of future simulators. There has been great difficulty in getting simulators high priority in aircraft programmes in the UK for many years. This is fortunately changing rapidly, but if simulators had been presented as being an integral and essential experience gathering tool supplementing the parent aircraft, and not labelled purely as a training aid, and requirement papers placed simulators on an equal footing with the aircraft, the case to fund suitable simulators would have been strengthened enormously. The need and value of dual training aircraft is seldom questioned, and yet they are limited devices in relation to our modern flight simulators.

MISSION REHEARSAL

At the beginning of this paper the Harrier GR5/7 simulator was described in some detail, and it was indicated that it will have a full mission rehearsal capability. It is interesting that the User is now accepting that missions can be rehearsed in specific instances. This in its own way provides a link in this thesis; high fidelity simulators that are capable of mission rehearsal, are machines that are as capable as front-line aeroplanes in everything but offensive or defensive effect, and must be experience building in themselves. This premise would be further supported if the German Air Force proposal to transfer all low flying into their advanced Tornado simulators is found to be feasible. At a lower training level, it should also be noted that the RAF is effectively conducting mission rehearsal in the Tucano simulators used as an integral part of new basic flying training patterns.

SUMMARY

In sum, those concerned with flight simulation have made great progress in developing machines to meet our training needs. However, the User's approach to the employment of advanced simulators is based on its origins as a training aid, when it has much greater utility. The User community should recognise that experience development is now a real proposition in advanced simulators, and quite apart from the practical benefits of employing them as such, the conceptual acceptance of simulators in that role opens up new avenues to explore. Possibly the greatest benefit would be the acceptance of flight simulators as an active flight safety tool. Cockpit Resource Management training development, simulator specifications, and aircrew self-tuition are but some of the other areas affected.

Many Users wonder whether the next breakthrough in simulation will occur in display systems, image generators, 'G' simulation, or some other field. It is possible that the next major breakthrough will occur when the Users of the modern simulators that advanced technology has provided, recognise them as having capabilities beyond "training".

REFERENCES

Mission Rehearsal - More than just another Simulator. R Wiggins Et Al. IITSC 1989.

Training Mental Rotational Skills. Damos. 18th WEAAP Conference.

Human Factors for Pilots. R Green Et Al. Averbury Technical Publications 1991.

Harrier GR5/7 Advanced Technology Mission Simulators. Clifford and Jackson. IITSC 1990.

Extracts of CHIRP reports. Green and Wilson. RAeS Advanced Cockpit Technology Conference 1991.

Initial Report on Acceptance of Tucano Simulators. MOD Crown Copyright.

The benefits of Flight Simulation. Marhon. Royal Aeronautical Society 1989.

Transfer of Training from Simulators to Operational equipment. Thompson. Journal of Educational Technology Systems. 1989.

Flight and Military Training and Simulation magazine articles.

ACKNOWLEDGEMENTS

The Author wishes to thank the many Industry and Military personnel who have given freely of their time briefing the capabilities of modern synthetic training technology, and demonstrating a very wide range of civil and military simulators. Numerous press reports, periodical articles, and other publications have provided the background for this Paper. Particular thanks are due to LTC Mark Russel, US Army, for his guidance and support.

THE AUTHOR

Wing Commander Bob Prothero holds the post of TO3(RAF) in the UK Ministry of Defence where he is responsible for the management of RAF flight simulators, and providing advice to the Policy Staffs on the application of synthetic training technology to future weapon system projects.

He is a graduate of the Royal Air Force College, rated as an exceptional pilot and instructor, and has held a wide range of flying and staff appointments. He is a member of the Royal Aeronautical Society, and British Institute of Management.

The Advanced Amphibious Assault Front End Analysis Process: An Approach to Balance Design and Ownership Requirements

Dr. David J. Daly, Mr. Mack Perry,
Dr. Charles A. Beagles,
Naval Training Systems Center
Orlando, Florida
and
LtCol James Feigley
U.S. Marine Corps

ABSTRACT

One of the primary goals of any major acquisition program is to achieve the best possible balance between performance, risk, schedule, and cost. Early consideration of life cycle cost and manpower, personnel and training (MPT) issues is critical to the achievement of this objective. Historically, operating and support (O&S) and MPT support requirements have not been adequately considered during the early phases of weapon system development. Consequently, O&S requirements have become the unaltered by-products of initial engineering decisions and in some cases have become a logistics support/MPT burden on the user community.

This paper presents one promising technique for the incorporation of O&S forecasts into the engineering requirements analysis process. This design to ownership approach requires concurrent and interdependent front end analysis. O&S predictions are generated by economic modeling of baseline and new system concepts. These early O&S forecasts lead to the generation of engineering design approaches and specific design rules to offset future support requirements.

INTRODUCTION

This concurrent engineering approach is being used for an acquisition category (ACAT) I program during the concept phase when historically engineering decisions account for 70% of a system's life cycle cost. The purpose of this paper is to explain this design to ownership process and to present some preliminary findings. The lessons learned from this acquisition should apply to both future weapon system and training system procurements.

The Management Problem. The critical issue to be addressed by program managers in today's declining budget environment is:

> With increasing weapon system complexity, how can we develop an *affordable* system which optimizes *total system* (man & machine) performance?

As indicated by Table 1, human integration issues are often bypassed by technical design and schedule priorities. As a result, system performance and O&S requirements have been significantly impacted.

The initial estimates of manpower (IEM) for complex weapon systems have often been significantly understated.
The maintenance demands for complex weapon systems have often been understated.
Operators have been required to remember too many steps to find targets and fire weapons.
System performance has been degraded by the demanding physical requirements placed on the user.
System performance has often been degraded because training of key collateral skills was not recognized as important.

Table 1. Past Ownership Problems

<u>The Program Manager's Concerns</u>. Marine Corps management recognized the magnitude of the potential ownership problem for the new vehicle, the Advanced Assault Amphibian Vehicle (AAAV). The AAA Program Manager requested assistance from NAVTRASYSCEN in analyzing the O&S and MPT requirements of the new system. Table 2 presents the program's manager's assessment of the situation.

Force Level Reductions are a Certainty.
Lessons Learned from the "Tech Base" show a high probability of greater technical complexity.
Maintenance of the Current System is taxing the Logistics and Training Support System.
New Systems do not necessarily eliminate "Old" problems.
Long term trends in Defense Spending will increase the emphasis on reducing the "Cost of Ownership" of New Systems.

Table 2. Program Concerns

NAVTRASYSCEN was included on the AAA team addressing AAAV procurement issues in 1988 prior to the initiation of the Concept Exploration Phase. This early involvement permitted the evolution of a systematic approach to the O&S problem. Tasks leading up to the concept study included:

- the implementation of training situation analyses which addressed the operational manpower, personnel and training constraints of the current system as well as current supportability problems.

- selection of an economic model to support conceptual design trade-off analysis.

- the development of a HARDMAN Supplement for the AAAV RFP which included development of three HARDMAN (Hardware vs Manpower) data item descriptions (DIDS).

- the development of an integrated front end analysis approach to include O&S/MPT considerations in the early design phases of the AAAV.

- the development of a Manpower, Personnel, Training, and Safety (MPTS) Plan for Defense Acquisition Board (DAB) Milestone Review.

THE DESIGN TO OWNERSHIP SOLUTION: CONCURRENT FRONT END ANALYSIS

The design and development of the new weapon had to fit the total affordability and supportability constraints of the Marine Corps. To address this goal, the AAAV RFP required that systems engineering, human integration, and logistics analysis be interdependent. As indicated by Figure 1, the selected approach also required the use of economic modeling to support that interdependency.

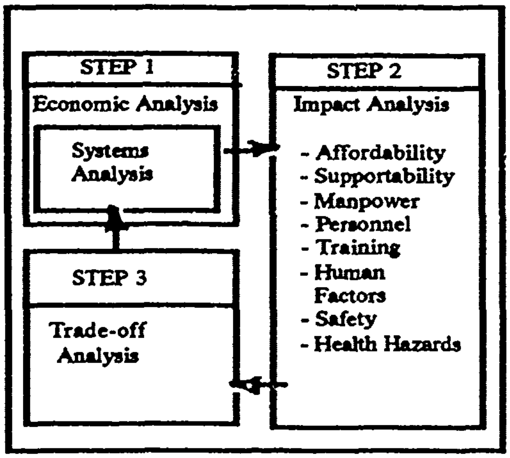


Figure 1. The Front End Process

Each step (Systems Analysis, Impact Analysis, and Trade-off Analysis) is presented in detail in the Army's HARDMAN Comparability Methodology. The AAA Program's addition to this process will be further described. The two key aspects of the approach which fostered interdependent design analysis are the use of past problems and economic modeling.

The Use of Predecessor and Generic Amphibious Problems.

As indicated by Table 3, new designs were assessed with regard to affordability and supportability, as well as by all the MANPRINT (i.e., manpower integration) domains.

For the AAAV analysis, past amphibious problems were input into the "Systems Analysis" because these problems drain O&S resources. Table 3 presents examples of the problems and lessons learned utilized from a review of training situation analyses, lessons learned reports, task listings and training device studies.

Industry's requirement was to analyze past user problems and to develop measurable design solutions. Each of the past problems identified were listed with an identification of potential design solutions. For instance, the need for an excessive amount of tools could potentially be reduced by requiring standard connectors in the new design. In this case, the requirement for a design to use "no more than seven tools" is a proposed design rule for further evaluation.

Logistics Number of Tools Required Amount of Test Equipment Required Time Required to Access Damaged Equipment
Manpower Number of People Required
Personnel Skill Levels Required
Training Sustainment of Gunnery Skills Sustainment of Troubleshooting Skills
Human Factors Complexity of Turret Operation Procedures Visibility when Buttoned Up
Safety Malfunction of Heater Unexpected Hatch Closure
Health Hazards Exhaust Fumes

Table 3. Past Vehicle Problems

Table 4 presents a list of design rules being evaluated by industry in their new design concepts. If a design rule adversely impacts the feasibility of the engineering concept, then a trade study or economic analysis can be run to support a decision on the design rule. The output of this analysis is a listing of potential design rules and an audit trail of problems and proposed solutions.

O&S Issue	Design Rule
Workload	
Tools	- No More than 7 Tools Required for Onboard Maintenance.
Test Equipment	- None Required for Onboard Maintenance.
Test Equipment	- No Growth of Test Equipment over Current System.
LRU Replacement	- All Lowest Replaceable Units (LRUs) of the same type shall be interchangeable. - All LRUs must be easily repaired and replaced while wearing NBC/Cold Weather Clothing. - All Fasteners must be Captive. - LRUs will be designed for replacement in uncontrolled (ie., moisture, dust, electrical) environments.
Accessibility	- There shall be no requirement to remove other equipment or parts to gain access to an LRU.
Adjustment	- Electrical adjustment will be automatic. - Mechanical Interface will require alignment features such as precision mounting surfaces and alignment guide pins. - Track adjustment will require no more than 2 people in 10 minutes.
Affordability	
Maintenance Ratio	- The Ratio of the Cumulative Number of Corrective and Preventative Maintenance Man-hours expended in Direct Labor and the Cumulative Number of End Item Operating Hours shall not exceed an 8 to 10 Ratio.
Human Factors	
Operating Procedures	- Operators will not have to remember more than 7 Steps in a sequence to perform any procedure.
Training	
Skill Sustainment	- Operators will be able to Perform Driving, Gunnery, and Repair & Replacement Procedures to 90% accuracy one month after training.
Safety & Health Hazards	
Crash Padding	- There shall be padding & back support for each Crew and Troop Position.
Hazardous Fumes	- There shall be No Fumes in the Crew and Troop Compartment.

Table 4. Typical Design Rules being Evaluated for Implementation

The use of Economic Analysis. Economic analysis was used to analyze the life cycle cost and MPT support requirements associated with the current and proposed vehicle. An initial objective was early identification of "high drivers." The intent was to translate these high drivers into engineering challenges and thus integrate O&S issues into the systems design process. The ultimate end product would be a more cost and operationally effective vehicle design.

An additional objective of the economic analysis was to support the trade-off analysis process by projecting the relative life cycle cost of alternative system designs. The Equipment Designer's Cost Analysis System (EDCAS) was selected as the economic model because of its sensitivity and trade-off analysis capabilities as well as its capability to project life cycle cost based upon preliminary design parameters. The use of the model will be described further in more detail.

Task 1: Economic Modeling to Determine where to Place Design Emphasis. Design contractors were initially required to model the current system, the Landing Vehicle Track-7A1, and a baseline system representing new technology. The resulting life cycle costs and manpower requirements were then analyzed by subsystem and ranked in terms of importance as demonstrated by the Pareto chart depicted in Figure 2.

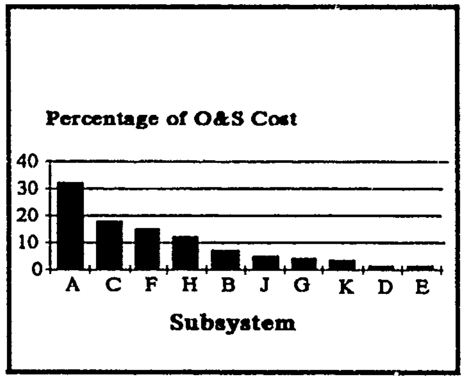


Figure 2. "High Driver" Subsystems

Using the results of the economic modeling, the engineering team could prioritize their design approach to minimize O&S impact. For example, the O&S requirements identified with the top four "high driver" subsystems in Figure 2 represent 65% of the total O&S cost and would naturally be given design priority.

Task 2: Economic Modeling to Support Trade-off Analysis. As the conceptual design matured, each subsystem was analyzed by varying the input design parameters (unit cost, scheduled maintenance, mean time between failure, mean time to repair, etc.,) to evaluate each design variable's impact on life cycle cost. Industry used this sensitivity analysis to identify the variables which had the greatest influence on life cycle cost. Figure 3 presents a "generic" representation of the relationship of design parameters to life cycle cost. Using the economic model in this context, logistics and MPT support requirements are quantifiable and thus are able to become an integral part of the trade-off decision process.

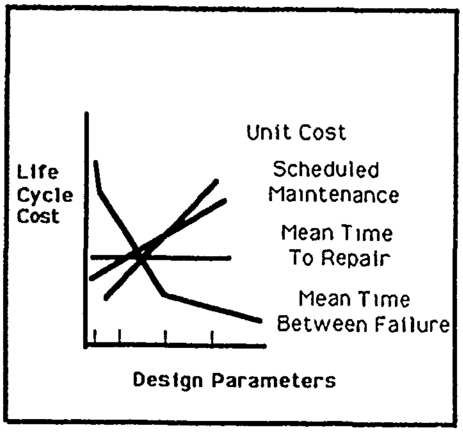


Figure 3. Sensitivity Analysis

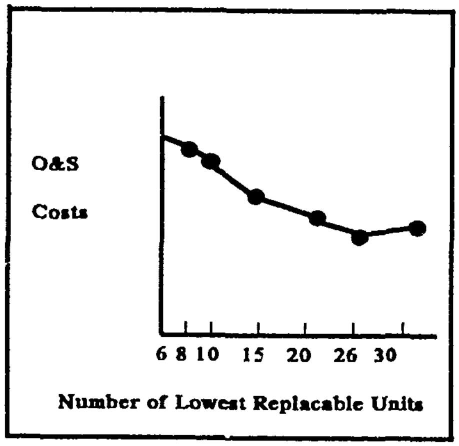


Figure 4. Partitioning Analysis

<u>Task 3: Economic Modeling to Support the Development of Modularization Goals</u>. In addition to sensitivity analysis, industry teams conducted partitioning analysis to determine the relative life cycle savings which could be obtained through modularization. From a supportability standpoint, it is often cheaper to break equipment in smaller and lighter lowest replaceable units (LRUs).

Figure 4 illustrates a generic partitioning analysis graph For the AAA Program, industry was required to develop a partitioning design goal for each subsystem design. In this case, the typical starting point would be for a design range from 20 to 26 LRUs. This would maximize the savings accrued through modularization. Next, a trade study would typically be run to determine what specific number of LRUs is feasible from an engineering perspective.

DISCUSSION

In the AAA Program, O&S impacts have influenced several key design decisions through the use of this type of concurrent engineering In some cases, a design alternative requiring the least expensive support requirement was selected. In other cases, the use of specific design goals and modularization goals are being used to reduce the ownership cost of "high driver" subsystems.

CONCLUSION

This process represents a major departure from past practices. For the AAA program, the economics of the new system's ownership costs are driving the design, instead of the design driving the ownership costs.

ABOUT THE AUTHORS

LTCOL JAMES M. FEIGLEY is currently the Assistant Direct Reporting Program Manager for the Advanced Amphibious Assault (AAA) Program. He has served as a platoon and company commander and battalion operations officer in the 1st Amphibian Tractor Battalion, 2nd Assault Amphibian Battalion, and 1st Tracked Vehicle Battalion. He is a graduate of the Amphibious Warfare School; Marine Corps Command and Staff College; Project Managers Course, Army Logistics Management Center, Fort Lee, Virginia; and the Program Managers Course, Defense Systems Management College, Fort Belvoir, Virginia.

DR. DAVID J. DALY is managing the Advanced Amphibious Assault HARDMAN Analysis. He has most recently been involved in industry with weapon system developments integrating economic, logistics, and HARDMAN analyses. He holds a Ph.D. degree from Indiana University in instructional systems technology.

MR. MACK PERRY is currently an Operations Research Analyst within the Performance Assessment Group of the Naval Training Systems Center (NTSC). He has fifteen years of experience with NTSC and PM TRADE conducting cost and economic analyses. He holds a M.S. degree from Rollins College Crummer School of Business.

DR. CHARLES A. BEAGLES is Deputy Program Director for the Special/Combined Operations Directorate, Naval Training Systems Center (NTSC). He has worked in training analysis and design since 1976. Prior to joining NTSC, he worked at the Center for Needs Assessment and Planning, Florida State University. Dr. Beagles is a former Marine Corps Captain and served as a Rifle Company Commander in Vietnam. He holds a Ph.D. degree from Florida State University in educational psychology.

AIR NATIONAL GUARD PART TASK TRAINERS
A Flexible, Cost-Effective Addition to Fighter Pilot Training

Major Brent Marler
National Guard Bureau
Requirements and Development Office
Washington, D.C.

ABSTRACT

Technology advances have made fighter aircraft cockpits increasingly complex, adding significantly to the requirements for training pilots in systems operation and "cockpit management" tasks. Fortunately, these same technology improvements have enabled new approaches to meeting these training challenges. In the Air National Guard (ANG), there are particular challenges associated with maintaining and honing the combat skills of traditional Guardsmen who serve part-time as pilots in operational units from Hawaii to Cape Cod.

To meet these particular needs, the ANG embarked on an acquisition program for training systems that capitalize on the improvements in computer technology, designed around low cost commercial systems capable of mission procedures training in a dynamic flight environment. The product - specialized trainers for specific procedures. These trainers are designed primarily for use by a single pilot, though there are provisions for an instructor. The flight simulation is not intended to match the high fidelity simulation levels found in current, full system simulators. However, the devices, called "Air National Guard Part Task Trainers" (ANG PTT) do provide a training capability previously unavailable to ANG pilots.

This paper describes the ANG PTT program from the first steps of evaluating available technology through requirements definition, Request for Proposal (RFP) development, source selection, and contract award. Applying lessons learned as a user of systems acquired through other agencies, the ANG designed this program along Total Quality Management (TQM) principles with support from Headquarters Air Force, Air Force Systems Command, Tactical Air Command, and industry.

INTRODUCTION

Recently, the nation was galvanized by radio and television as the drama of Desert Shield and Desert Storm unfolded a "prime time spectacular." Video images of this stunning victory, led by airpower, captured the attention of countless Americans. Scenes of "smart munitions" dropping down smoke stacks along with Head-Up Display (HUD) video showing kills of Iraqi fighters demonstrated the technological capabilities of the United States to the world. Yet, these images carried a critical, if unseen message - technology without proper training results in a paper tiger. There is no doubt that the superiority of American warfighting technology was convincingly demonstrated. However, the Iraqi forces also had some of the best equipment available both from the East and the West. To accept that their defeat was solely due to U.S. Technology superiority would clearly ignore the inability of the enemy to employ their purchased modern technology.

The lessons of Desert Storm validate and reinforce the emphasis the ANG has placed on training for several years. Unit conversions to the F-15 and F-16 continue at an accelerated pace. These aircraft, and other current USAF fighters, rely increasingly on Hands On Stick And Throttle (HOTAS) controls for systems operation and mission performance. Even experienced pilots need frequent practice and refresher training as they transition to these modern systems.

With these concerns for training as the basis, the National Guard Bureau (NGB) Requirements and Development Office developed a strategy for acquisition of F-15 and F-16 Part Task Trainers specifically designed for use by mission ready pilots in ANG operational flying units. This led to award of a contract in September 1990. This event is a major milestone in the ANG's first in-house acquisition program for flight training devices.

The program applies off-the-shelf technology to meet the user's system performance, cost, configuration concurrency, and reliability goals. Another important program goal was to apply Total Quality Management (TQM) principles in defining user requirements and working with industry to find solutions and alternatives to meet them.

The paper also discusses system architecture and software programming approaches that best support the variety of aircraft configurations and scenario designs needed by different units. Conclusions and projections are included.

WHAT IS A PART TASK TRAINER?

Part Task Trainer (PTT) - what is it really? Terminology in current Air Force regulations has not been able to keep up with the rapid changes in trainer technology. Typically, PTTs are viewed as training one procedure only or are used for preparatory training for high fidelity simulators in student training scenarios. The concept of a PTT capable of air-to-air, air-to-ground, instruments, and emergency procedures doesn't fit the stereotypical view, but neither does the training technology now available.

Are we going backward? Visitors to the Training Systems SPO at Wright-Patterson, AFB, OH, are greeted by a beautifully preserved Link trainer - the primary ground training device used during WW II. This trainer, built in the form of a miniature airplane with wings, tail and cockpit, is not unlike the PTT of today. It's shape communicates a sense of entering an aircraft cockpit. The instruments look and function as in the real airplane. Its impact on pilot training was significant. It's descendants have become marvels of technology, providing training capabilities unheard of not many years ago. Certainly, these high fidelity simulators will dominate flight training for many years to come for they provide training capabilities far beyond those of low cost systems such as the PTT. It is, therefore, not the intention of this paper to advocate their replacement. On the contrary, the National Guard Bureau has established firm requirements for such advanced systems for our schoolhouses - our F-16A, F-16 ADF, F-16C, and RF-4C Replacement Training Units.

Conversely, the ANG PTT acquisition focus was on the word part in Part Task Trainer - specifically the part relating to procedures. We want to refresh pilot skills in the procedures required for several specific mission tasks. We need to make this training easily obtainable and effective. Targeting procedure training reduces simulation fidelity requirements permitting the use of part task trainers instead of full fidelity simulators - and, not insignificantly lowering, cost. As with everything else in life, this focus is not without its own challenges. For example, how do you define the limits of the required fidelity? This question was raised by several industry representatives at the PTT pre-solicitation bidders conference in October of 1989. An adequate answer to this question is important to the government and to potential bidders. Contractor's see "fidelity" as a cost issue - a direct relationship that is critical to bid and proposal preparation. The government needs to articulate the ANG F-16 Conversions level of fidelity desired as clearly and concisely as possible in each area of training. It is not an easy process. Perhaps the challenge in adequately defining the required fidelity stems from the newness of these high technology-low cost devices.

FY 92		
192	TFG	Richmond, VA
181	TFG	Terre Haute, IN
122	TFW	Fort Wayne, IN
114	TFG	Sioux Falls, SD
140	TFW	Buckley ANGB, CO
185	TFG	Sioux City, IA
182	TFW	Peoria, IL
180	TFG	Toledo, OH
104	TFG	Westfield, MA
156	TFG	San Juan, P.R.
FY 93		
178	TFG	Springfield, OH
128	TFW	Madison, WI
103	TFG	Bradley ANGB, CT
132	TFW	Des Moines, IA
138	TFG	Tulsa, OK

Table 1

DEVELOPMENT OF USER REQUIREMENTS

Who is the User?

A look at the ANG F-16 unit conversion schedule (Table 1) makes it apparent that installing full system simulators at each location would not be a cost effective means to meet training requirements, even if enough simulators were available from the USAF - the customary source of ANG equipment. The organization of the ANG contributes significantly to this. By necessity, units need to be located in areas that can support recruitment of part-time airmen. This situation results in a basing plan that works for the ANG, but does not afford the economies of scale seen in the USAF Wing structure.

The typical ANG flying unit, designed as a Group or Wing, is composed of a single flying squadron and support organizations. The flying unit will usually have 18 or 24 primary assigned aircraft (PAA). The unit supports approximately 40 pilots, including squadron mission ready pilots and attached pilots from the group/wing. The unit location may be in a large metropolitan area, like the 113th Tactical Fighter Wing (TFW) in Washington, D.C., or it may be located in a rural area in South Carolina or on Cape Cod in Massachusetts. The diversity of location means that long commute distances from home to unit or to the nearest available simulator are a consideration in designing unit training programs. The former affects the availability of pilot time at the unit. The latter can limit time available for the most effective type of training - flying time in the cockpit of unit aircraft.

Request For Information

In the process of evaluating ANG unit training requirements, it became clear that improvements in training system technology could provide a new type of

trainer that would support effective training at each unit, reduce non-productive time spent in traveling to the training site, and fit well with the availability profiles of traditional Guardsmen. Such a trainer would need to provide skill refresher training to experienced pilots who may have low time in the current aircraft, but are maintaining mission ready status at the same levels required of active duty pilots. The areas of interest are systems operation in air-to-air and air-to-ground, instruments, and, for the F-16, engine emergency procedures. In August 1988, the ANG published a Request for Information (RFI) in the Commerce Business Daily (CBD) asking for the views of industry on the possible use of ground training devices for current and future requirements. Contractors were invited to submit, at no cost to the government, data on a training device or devices that could meet the ground training needs of pilots at operational F-16A/B units. The notice garnered 26 responses from a wide variety of firms in the training and simulation business. Together, the responses provided an excellent survey of industry capabilities, projections and recommendations that the ANG used to begin developing an acquisition program.

PTT Requirements Defined

The number and quality of responses to the RFI indicated that the goals identified for a unit training device were feasible. In the summer of 1989, the development of the Request for Proposal (RFP) began in earnest. Critical to the requirements definition process was involvement of the user. The basic requirement was to provide a trainer for mission ready ANG pilots. Pilot availability considerations drove the requirement for the trainer to be usable by one pilot operating alone. This led to a requirement for the trainer to include a feedback mechanism - a Performance Feedback System (PFS), so that a pilot could obtain a performance assessment in the absence of an instructor or qualified observer. To enhance the capability of the trainer, an instructor station was also required. The training provided was to be effective for pilots new to the aircraft, as well as for "old heads" - a challenge of no small magnitude. The device was to provide air-to-air training from beyond initial detection to the minimum range for the aircraft's shortest range missile. Procedure training in instrument cross check was also desired features, as was engine airstart emergencies and air-to-ground procedure training for the F-16. Finally, the capability to network trainers together was added, primarily to ensure the capability for future growth. The ultimate goal was to provide a trainer that was effective, easy to use and always available, even for pilots with only a few minutes available time.

Control of cost was a major consideration. The NGB had an established budget for the acquisition. Our target was $500K per system - total cost. Since the information we had indicated that the requirements beyond air-to-air might drive the cost beyond our budget, NGB developed its RFP with specific capabilities defined in option packages, each to be priced separately. Also, the device was to be unclassified - security requirements for classified would increase costs and decrease the ease of access. In this way the capability of the system acquired could be tailored to fit the budget. Thus, the requirements for engine airstarts, instruments, and air-to-ground were packaged as contract options. This approach allowed procurement of maximum capability with the available funds. It also permitted elimination of options that were viewed as "over priced." Finally, we wanted the best trainer for the F-15 and the best for the F-16. To insure that we could do this, offerors were permitted to bid either or both aircraft, thus, permitting the government to acquire the PTTs from different offerors.

At this time the ANG had several F-16s units without simulators and was anticipating many future conversions to the F-16 (Table 1). Since available funding was limited, the buy was split into two phases - phase I to meet the immediate near term need and phase II to meet the requirements for FY 92 and beyond (Table 2).

ANG PTT LOCATIONS

F-16

113 TFW	Andrews AFB, MD
127 TFW	Selfridge ANGB, MI
149 TFG	Kelly AFB, TX
169 TFG	McEntire ANGB, SC
174 TFW	Syracuse, NY
183 TFG	Springfield, IL
187 TFG	Montgomery, AL
188 TFG	Fort Smith, AK

F-15

102 FIW	Otis ANGB, MA
116 FIW	Dobbins AFB, GA
131 TFW	St Louis, MO
142 FIG	Portland, OR
154 Comp Gp	Hickam, HI
159 TFG	NAS New Orleans, LA

Table 2

Pilots were included in the requirements development process early on. As the ultimate user of what we would buy, their involvement was needed and actively solicited. This led to many lengthy discussions on how best to define the requirements, but it also led the program office to a better understanding of what would make an effective trainer.

One primary program goal was to provide a trainer that pilots would want

to use, so factors perceived to positively affect pilot acceptance were included in the requirements. The fidelity (form & function) of the controls for the primary areas of training, i.e., the stick and throttle, were of obvious importance. The realism of scenarios, ease of scenario design, and user friendliness were also important for user acceptance as well as training effectiveness.

Fidelity

As previously mentioned, fidelity is important - it is also a challenge. It must be high for the primary areas of training, but, in other areas costs can be reduced by lower fidelity - provided it does not detract from the primary training. An aero model for a PTT may not require edge of the envelope flight performance, but it must be good enough to "fly" the PTT straight and level so the pilot can concentrate on learning or improving a cockpit task. If the pilot spends all of his time trying to keep the wings level or maintaining altitude, he is not learning the procedures for which the PTT is designed.

The fidelity required for each PTT task should be carefully defined. In our PTT, HOTAS controls are critical to the desired training, therefore, they require high fidelity. Conversely, switches and knobs unnecessary for the procedures being trained required no fidelity beyond a two-dimensional representation. In all of these things, the goal was to provide quality training - training like the paircraft. This really translated to avoiding things that would require pilots to learn functions or "feel" different than the aircraft. For that reason the use of the HOTAS controls for user interface with the PTT presents concerns.

One of the challenges in judging fidelity is the affect of human perception as illustrated in the following story. It seems that a new stop light was installed at an intersection used each day by two city councilmen on their way to and from work - one passed through it going north and south, the other going east and west. Both complained to the city traffic engineer that the "red" light was too long. Perplexed, the engineer did nothing. Several days later he telephoned each councilman telling them that the light had been adjusted some days before and asking how they felt about the light now. Each councilman replied that it was now fine.

How well the PTT HUD looks in comparison to the aircraft is very important. If a MIL SPEC HUD were used one would probably perceive that it's display is just like the airplane. However, when someone is asked if the computer generated HUD display on the PTT is like the airplane, they are likely to look more closely to specific details than they do in the airplane. The resulting comments may be completely accurate, but they may also reflect individual perception. We have had pilots fly the prototype PTTs from time to time. Some would say dots on the HUD were too large or the horizon line is too long. The fundamental issue is that the systems in the PTT must be documented from the Technical Orders and then carefully balanced with user input to provide proper perception.

In keeping with the concept of user acceptance, consideration to the user interface is very important. A menu and scenario design system must, therefore, be designed for the non-computer experienced user. This should be standard practice today. However, designing a simple and effective user interface is not necessarily easy, but failure to do so can inhibit pilot use. Of equal importance is the speed of system boot up, loading a new scenario, and accessing performance feedback, etc. No one likes to wait for things to happen - especially pilots.

ANG PTT CAPABILITIES

The PTT consists of a replication of the aircraft cockpit and a color out-the-window (OTW) field-of-view (FOV) that includes the HUD display. To control cost, the switches, gauges, etc. are two dimensional unless they are necessary for training. Upon turn-on (or if selected by the operator) the PTT will display on the versions of the aircraft OFP and PTT software currently loaded. This is done to avoid confusion when a units aircraft under go an OFP change since the PTT may be upgraded prior to the aircraft. The OTW visual display uses Defense Mapping Agency (DMA) Digital Terrain Elevation Data (DTED). The DTED terrain is computer "textured" to enhance depth perception and realism. This improves the ability to

Prototype of the ANG PTT showing the cockpit and Instructor/Operator Station (IOS). (U.S. Gov't photo)

maneuver at low altitudes and overall pilot acceptance. The area of terrain provided includes desert, mountains, coastal plain and sea.

The PTT is set-up using a touch screen monitor on the Instructor Operator Station (IOS). User friendly menus permit the user to select a scenario then fly, review, and save it, as desired. The instructor has password accessible menus for designing the scenarios and selecting the Performance Feedback System (PFS) parameters desired for the mission. The PFS parameters can be individually "weighted" to emphasize the particular training desired. The IOS includes a keyboard and joy stick in addition to a monitor. An interesting IOS capability is that a second pilot can control one of the PTT generated aircraft and fly it - as a wingman or an adversary - using the joy stick and keyboard.

For air-to-air training, the PTT can provide up to ten individual targets. These targets include fighters (red & blue), bombers, small airplanes, helicopters, and surface-to-air threats (both missile & anti-aircraft artillery). They can be arranged as desired by the scenario designer. For example, a scenario can be designed with ingressing ground attack aircraft escorted by two air superiority fighters. This combined with the PFS permits units to tailor training to mission taskings, address trends identified in check flights, or emphasize particular desired habit patterns. A Radar Warning Receiver (RWR) and chaff/flare dispense system are part of the PTT.

Limited procedure training in the major bombing modes - specifically Visual Reference Point (VRP), Visual Initial Point (VIP), etc. - is provided for the F-16. For this, ground targets and navigation points are provided that can be arranged using the IOS scenario design. The system concept is to provide HOTAS training inbound to the target - not to fly low levels. To enhance this training, random drift errors are induced requiring pilot actions to correct. Since the software is fully integrated, a scenario can be designed where the PTT is "attacked" - by surface or airborne threats - at anytime during air-to-ground training.

A limited instrument capability is also included for practice of ILS and TACAN approaches. The system is generic, i.e., the scenario designer can input approach course and field altitude to simulate any airfield. Even the heading is changed on the end of the runway. Thus, approaches can be designed for deployments or home station training. No PFS is provided for instruments.

The capability to practice both normal and emergency airstart procedures is included in the F-16 PTT. The emphasis is on training the procedures, not on modeling the flight envelope. This is included in the PFS since the PTT is designed for solo practice.

Overall, we were able to acquire more capability per dollar spent than anticipated from the RFI results. Some, probably most, of that is certainly due to the benefits of a competitive acquisition, but some saving assuredly results from the rapid and continuing advances in technology. The latter consistently provides greater capability each year while costs continue to decline. The lesson, you can control your costs by using contract options, but you will probably be able to acquire more capability than your market surveys will indicate - maybe you really can "do more with less!"

SYSTEM MANAGEMENT

System Maintenance

The ANG recognized the importance of having a trainer that was available and functioning properly whenever a pilot wants to use it - a system that is not functioning properly is likely to be avoided. Additionally, full time personnel at ANG units are few, so it was desirable to minimize the tasking the PTT would place on the unit. For these reasons, special attention was given to system reliability and support. A full Contractor Logistics Support (CLS) package was procured with the system, as a result. The PTT is required to be available for training at a 95% rate based upon use time. The simplicity of the system permitted the government to stipulate that breaks are all or nothing - that is, if any part of the system is malfunctioning, the whole system is considered down when calculating the availability rate.

To control costs with PTTs spread across the country, NGB desired a trainer that did not require on-site maintenance personnel. The result was a RFP that allowed a maximum of three days to repair any malfunction. For this concept to work, the contractor must be totally responsible for the repair of all items in the trainer and not rely on government depot repair. For ease of reporting system problems the CLS contractor was required to have a toll free 800 number. Since the PTT will be used days, evenings, and drill weekends, the contractor must provide for 24 hour, seven days per week reporting. In addition, the three day repair period starts when a problem is reported and is counted according to the units work schedule, including weekends. Prompt repair to insure system availability has a direct impact on user acceptance. With this in mind, modem connection to each PTT is provided to permit remote checking of system performance and trouble shooting.

System architecture and software

System support was also addressed in the context of keeping the trainer configuration current with the aircraft configuration. The contractor is required to monitor all hardware and software changes to the aircraft. When changes that will affect the PTT are identified, the government CLS manager is to be notified for approval to update the PTT. The most critical part of the contractor's responsibility is to field the PTT update no later than the fielding of the update to the aircraft.

One aspect of meeting the requirement to maintain the PTT concurrent with the aircraft configuration is the system design. System concurrency updates can affect hardware as well as software since configuration updates may add new switches, controls, etc. An important aspect of a PTT's software design is the ease of software updates. It appears that using object oriented languages can simplify configuration updates and, in turn, reduce costs and update time. However it is done, the flexibility to accomodate hardware & software updates must be designed into the system.

FUTURE REQUIREMENTS

Continuing advances in training systems technology will permit future PTTs to have more capability than those today. The capacity to take advantage of these advances is important if a PTT is to remain a viable trainer for six, eight or ten years. The difficulty is in predicting what capabilities those advances will offer and what capability they may add to the PTT. However, here is an attempt to look ahead based upon capabilities that are in work, areas that need research, or may become future user requirements. This is in no way a determination that the items below can be effectively trained in a PTT like trainer - that must await future evaluation.

Fidelity. Better low cost sticks and throttles are required. This is true for most "hands-on" items in the cockpit. Expanding to full functionality such items as the F-16 Stores Management System (SMS) and Fire Control Navigation Panel (FCNP) would offer greater training capability though all functions may not be required. Along with these "hand-on" issues is the need for better low-end simulation software for things like the aero, missiles, ECM, and threat models. It would really be marvelous to design scenarios to allow training against an adversary's tactics - things like enemy fighters trying to drag the PTT into the enemy's SAM envelope.

Networking. This is a capability that has received much attention over the past few years. It has the potential to greatly enhance training for many training systems including part task trainers. However, to be viable it must focus on improving training for the PTT pilot. There two approaches to networking - local and long haul. The ANG PTT takes advantage of the first through the use of it's "flyable" Instructor Operator Station (IOS). The instructor (or another pilot) can choose a mode that permits him to select and take control of one of the computer generated aircraft as a "man-in-the-loop" adversary or as a wingman for the PTT - though keeping track of a wingman is limited by the single channel visual. The ANG PTT is capable of long haul networking, but questions regarding the training to be accomplished must be answered first.

Full Field of Regard Visual. One improvement that would greatly enhance the PTT is a low cost system that will provide a full field of regard visual. With such a system the benefits of local networking would dramatically increase since it would permit realistic two-ship training - you could see your wingman! The development of low cost dome visuals may answer this need as may Helmet Mounted Displays (HMDs). The challenge is to find such a visual system at a cost reasonable as an addition to a $500K trainer and that will not require construction costs to raise ceilings, etc.

Use of Classified Data. Future PTTs will need to be designed for handling classified data if they are to meet the training requirements for new aircraft capabilities now entering the ANG. To continue to provide readily available training, these trainers will need a system architecture designed for ease of use, perhaps using a single classified disk. This would preserve the desired user friendliness since removing and installing a single disk would make it easy for pilots to use the PTT. It is also necessary to avoid the costs associated with the physical security required for open classified storage. Perhaps in the near future, optical disks will be able to simplify loading and unloading classified data. Consideration should also be given to providing an unclassified disk as well.

Emergency Procedures. The ANG PTT was designed primarily for use by one person. This presents challenges when training emergency procedures since no instructor is present to provide feedback. In addition, modeling all the possible errors a pilot may make in practicing emergency procedures can greatly increase the complexity and, therefore, the cost of the PTT - potentially removing it from the realm of low cost. Some of the cost associated with including emergency procedures can be controlled by carefully determining the specific emergencies that require training. Once that has been determined, the problem of keeping the PTT current with changes to the emergency procedures in the technical orders must be addressed. This becomes even more

critical if a performance feedback system is provided to compensate for the absence of an instructor.

Visual Scene. The capability to use Defense Mapping Agency (DMA) Digital Terrain Elevation Data (DTED) with computer texturing creates a visual scene much more realistic than polygon images alone. It also presents challenges beginning with the amount of processing required to keep the OTW at an acceptable refresh rate and continuing to the data storage required for scenario replay. Image generators are getting better every year providing greater and greater capability. This capability needs to be exploited to the fullest extent to enhance training realism. The use of DTED has the potential to allow training over any terrain. In the case of the ANG, it would permit pilots to train for their deployment theater of operations while at home station. This will be of particular value when Landsat or SPOT satellite imagery can be over laid on full resolution (100 meter) DTED with free fly throughout at least one geo-ref cell (1 degree by 1 degree). However, in order to keep this capability at a low cost, a common data base is needed. Project 2851 is an attempt to standardize the data base for simulation, but it has yet to be demonstrated on a low cost device like the PTT.

Other. Programs are in progress that have the potential to require Flir/Night and Close Air Support (CAS) training for ANG pilots. Along with those are other potential training needs for multi-ship tactics, network training of pilots and AWACs/GCI,and FAC to fighter operations. Whether or not these tasks can or should be trained in a PTT is an open question. That pilots will need continuation training for these tasks is not. Most flying squadrons do not have the room for multiple training systems, so using an existing trainer may be desirable. However, there is nothing to preclude adding a Computer Based Instruction System (CBits) to a system like the PTT.

The computer has greatly expanded the variety of ways that we can train people. The challenge is to take advantage of that and develop new and better ways to meet our training needs.

CONCLUSION

As with any program, we have learned how to do some things better, but we have also raised more questions. Some of these questions relate to technology and research and may not be answered soon. However, for these systems to improve answers are needed. Simply stated, how does one measure pilot performance in a PTT? Along with that, how should the feedback on that performance be presented to provide the best learning and increase the pilot's motivation to train more? These questions become more complex as PTT capabilities expand into aircraft emergencies, more systems functions and more complex scenarios. However, they go the heart of the issue - providing the best training possible in a low cost, solo operated trainer.

Furthermore, how should PTT capabilities be enhanced? What capabilities can be added simplest and best? What capabilities enhance training giving more "bang for the buck"? Ask most anyone for an opinion. Some want networking, but is it really viable with limited visual and the complexity of coordinating long haul training? Others want more emergency procedures, but what emergencies and how best to train them? Still others want expanded visuals for air-to-ground tactical applications; or AWACs, GCI, FAC, ECM training; and the list continues. However, we must be careful and pragmatic and not try to walk faster that we - or technology - can run.

One day, technology will allow us to make small, low-cost trainers that will provide very complex high fidelity training perhaps approaching todays simulators. That day is not today. Therefore, we should not focus on attempting to build such devices, but on making what we have better and carefully improving them as technology advances permit.

So the paramount question seems to be, "How do we do this better?" That seems to be the fundamental question many besides the ANG are asking about training, period. We believe that the PTT, like the Air Intercept Trainer (AIT) before it, is, to quote George Orwell, the beginning of "newthink" in aircrew training. We look forward with anticipation to the training benefits these and future low cost PTTs will provide. The ultimate goal is to keep our pilots the best in the world!

ABOUT THE AUTHOR

Major Brent W. Marler has serves as the Chief, Armaments & Avionics Branch in the NGB Requirements and Development Office (RD) of NGB and has served as the Program Manager for the F-15/F-16 Part Task Trainer (PTT) since program inception in 1988. A native of California, his home unit is the 163 TFG, California Air National Guard at March AFB where he was an Instructor Weapon Systems Officer, Flight Examiner in the F-4C & F-4E and chief of simulator training. Prior to joining the ANG, Major Marler was on active duty with the USAF serving tours at Kunsan, Korea and at Spangdahlem AB, GE before returning to Luke AFB, AZ as an F-4 instructor. He has a M.S. in Management from the University of Utah and a B. A. in History from Brigham Young University.

INTEGRATING A FORCE-LEVEL SIMULATION SYSTEM INTO SHIPBOARD COMBAT SYSTEMS

Barry B. Morton
Computer Sciences Corporation
304 West Route 38 P.O. Box N
Moorestown, NJ 08057

ABSTRACT

This paper discusses the goals, challenges, and lessons learned from integrating an established shore-based, force-level simulation system into the shipboard combat system environment. The results of this effort were demonstrated at the Fleet Combat Training Center (FCTC), Pacific, and the Tactical Training Group, Pacific (TACTRAGRUPAC) in May 1990. Real training platforms included Aegis, Spruance, and Perry class ship mockups. Simulated platforms included enemy and friendly submarines, fixed and rotary aircraft, and surface vessels all operating in a war-at-sea scenario.

INTRODUCTION

The primary goal of this effort was to demonstrate technologies available through the integration of off-the-shelf hardware and existing simulation and embedded training software for the enhancement of multi-platform, multi-warfare battle group (BG) training. The enhancements were designed to increase the realism of the existing training environment and to decrease the manpower needed to prepare and coordinate the training.

This project was accomplished under the auspices of the Chief of Naval Operations, Tactical Readiness Division (OP-73) and the Commander Space and Naval Warfare Systems Command (PMW-161). The technology demonstration consisted of interfacing an Enhanced Naval Warfare Gaming System (ENWGS) host computer, located at Tactical Training Group, Pacific, to a combination of shipboard combat systems. ENWGS is a multi-warfare command, control, computer, communications, and intelligence (C^4I) simulator of the entire spectrum of naval operations. It provides BG and amphibious group (AG) training support at the Tactical Training Groups and Amphibious Schools on both coasts and strategic wargaming support at the Naval War College. ENWGS is also installed at the Joint Warfare Center (JWC), Hurlbert Field, Florida. The combat systems were ship combat information center (CIC) mockups located at the FCTC, Pacific. Communication between the ENWGS host computer and mockups consisted of secured communications links via STU III over common telephone lines.

METHODOLOGY

This project first reviewed the "WHAT," "WHO," and "HOW" characteristics and technologies of the BG C^4I environment. Next, a concept of operations was developed to facilitate the long-term enhancement of the BG training environment. To minimize the cost of demonstrating some of these concepts, an existing simulation driver, ENWGS, was integrated with several shipboard combat systems. The integration permitted the ships' CIC teams to conduct a multi-warfare, war-at-sea exercise.

REVIEWING BG CHARACTERISTICS - THE WHAT, WHO, AND HOW

The WHAT characteristics of BGs consist of one aircraft carrier and from six to ten support ships, including one or more cruisers, destroyers, frigates, and logistic ships. A BG also includes organic fixed-wing and rotary aircraft, land-based aircraft, and often a supporting submarine. BGs are multi-warfare capable: they conduct war above, on, and below the surface of the water and, to a considerable extent, over land. A BG's geographic "area of concern" is generally theater-oriented, such as the Northwest Pacific or Persian Gulf. The time line for BG decision-making consists of strategic planning (measured in days), normal operations (measured in hours), and crisis management (measured in minutes).

The WHO, or the people who operate the WHAT, consist of fleet commanders-in-chief (and their staffs) in their shore-based command centers, a number of fleet commanders and battle force/group commanders in their flagships, composite of warfare commander (CWC) teams, which may be dispersed throughout the BG in ships, and "ownship" (each individual ship's) decision-makers (i.e., pilots, commanding officers, the CIC team, and individual equipment operators). They all receive decision-making information via consoles and group displays. Inasmuch as these people participate in the real-world operational environment, they must be incorporated into BG training, whether their involvement is simulated, real, or a combination of the two.

HOW do the WHO function in the WHAT? This question can be answered by understanding BG information flows from which the WHO extract information and make decisions. Information flows are generated via ownship information, or information that is either organic or non-organic to the BG.

Ownship information is usually generated by a platform's sensors, which might be a lookout or a ship's, aircraft's, or submarine's radar or sonar. Both humans and the ship's combat system process detections in an attempt to classify and understand the detection's composition and intent. Combat systems present processed detections to humans through equipment consoles and both individual and group displays. Equipment consoles consist of radar repeaters, Naval Tactical Data Systems (NTDSs), or sonar or electronic support measure (ESM) displays. Since most information presented to humans is in digital format, it is a fairly straightforward procedure to inject digitally generated simulation data, such as detections, into these consoles and displays. When sitting at an Aegis ship's command center console, it is difficult, if not impossible, to distinguish simulated contacts from real. The challenge is to present sufficient information to make each BG ship's CIC team member believe he or she is participating in a multi-platform, multi-warfare BG operation.

Ownship detection and resultant processing information is transmitted to other platforms by several means, the most typical being the NTDS. This type of information can be considered organic since it is within the BG's means of control. Organic information is of immediate tactical significance and is digitally displayed on ownship consoles and individual and group displays. Since NTDS is digitally communicated, it is fairly easy to simulate. The challenge is to include every NTDS platform, both simulated and real, in the exercise.

Information that the BG needs to support planning, but which originates outside its means of control, is considered non-organic. This information is strategic in nature and derived from intelligence sources such as national sensors (satellites) and underwater passive collection devices (e.g., SOSUS). This information is also displayed on ownship consoles and individual and group displays. The challenge to the training community is to interactively simulate non-organic events.

OUR PROJECT OBJECTIVES

Based on an understanding of the BG characteristics and information flows, and for the purposes of this project, we assumed long- and short-term objectives. The long-term objective was to provide "appropriate" information and data to "realistically" simulate/ stimulate BG characteristics (WHAT), through information flows (HOW) to the designated people (WHO) — anywhere, anytime. In order to manage the integration effort undertaken by this project, short-term objectives were to (1) increase realism beyond current training, (2) understand how much realism is necessary, (3) include as many functions, persons, and shipboard equipment as economically feasible, (4) decrease personnel overhead, and most importantly, (5) *keep the long-term objective in mind.*

THE INTEGRATION

The integration effort had several design constraints which affected our approach. These design constraints were to (1) make no modifications to existing combat systems; (2) make no modifications to existing shipboard simulation/stimulation devices; (3) limit shipboard carry-on simulation equipment and necessary connections to minimize installation/deinstallation and space requirements; and (4) limit the integration effort to ships in port.

The design approach consisted of making slight modifications to ENWGS and integrating it into several different classes of shipboard combat systems. (Refer to Figure 1.)

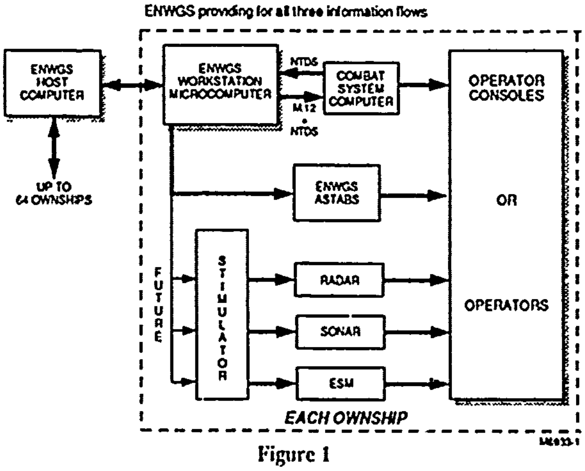


Figure 1

The point of entry into a ship's combat system was a hardwire connection into its NTDS data terminal set. Testing and final demonstration of the integration took place using the Aegis Training Center's (ATC) mockup at Dahlgren, simulating the USS Ticonderoga, an Aegis class cruiser, and two FCTC, Pacific mockups simulating the USS Spruance, a destroyer, and the USS Perry, a fast frigate. Those three "ships" were the only "live" platforms in the scenario. However, they found themselves participating in BG operations with the USS Vinson, an aircraft carrier, and a combat air patrol (CAP) and early warning aircraft. In direct support of the BG was the submarine USS Los Angeles. These friendly forces were all simulated within ENWGS. Hostile activity consisted of enemy bombers, cruise missiles, and submarines simulated by ENWGS.

Results of the integration effort satisfied, to a great extent, all of the BG information flows.

Ownship information was provided via several methods. Each mockup received its own active sensor detections from ENWGS by modeling the ship in ENWGS. When the modeled ship (e.g., the Ticonderoga) made a detection in the host computer, it was converted into an NTDS training message and injected into the ATC mockup's NTDS program. Each real ship received only detections made by the corresponding modeled ship. Thus, the Spruance could not "see" the Ticonderoga's detection unless the modeled Spruance also made the detection — that is, if the Spruance had its radar on and was within detection range of the contact. Detections were displayed on the ship's consoles and group displays as NTDS symbols. Humans could process these contacts as though they were actual detections. Visual detections via lookouts were automatically generated and portrayed on automated status boards (ASTABs). Passive contacts, such as passive sonar and electronic sensors indigenous to each ship, were also portrayed on ASTABs. Moreover, each mockup had the capability to maneuver, activate sensors, and fire weapons.

BG organic information (i.e., NTDS) was provided in several ways. NTDS information was generated by each mockup and communicated to every other mockup. In addition, every NTDS-capable simulated platform participated on the NTDS network. For instance, the simulated E-2C, an early warning aircraft, which was "launched" from the Vinson is, in real life, NTDS-capable. In our scenario, the E-2C was radar activated and was able to detect approaching air contacts; in our scenario, this was a hostile bomber raid. This detection was converted to an NTDS air track and transmitted to the BG in NTDS format, as it would be in real life. When the hostile bombers were radar activated, the ENWGS models detected this on their electronic warfare (EW) equipment. This was displayed to each ownship on its EW ASTAB. All NTDS information, as well as each ownship's contacts, was displayed on each mockup's consoles and individual and group displays. Humans determined that the EW detections corresponded to the approaching bombers, and the BG ships took appropriate actions to defend themselves.

Non-organic information was provided via several methods. The ENWGS-to-Naval Tactical Command System (NTCS – formerly JOTS) interface was one method and ENWGS ASTABS another. Information provided to the BG included SOSUS, High Frequency Direction Finding (HFDF), and satellite reports, all automatically and interactively generated by ENWGS models.

The demonstration that was arranged permitted free-play exercise. The scenario included a hostile bomber raid with anti-ship cruise missiles being engaged by CAP and a hostile

submarine prosecution with Spruance's anti-submarine warfare (ASW) LAMPS helicopter. During the course of the scenario, several hostile cruise missiles penetrated the BG defense, resulting in simulated damage to the Ticonderoga. The battle damage report generated by ENWGS informed the Ticonderoga crew (via ASTAB) that the forward missile battery and its EW sensors were out of action. This damage would affect the ship's fighting ability until repaired or until the exercise coordinator artificially intervened. The exercise did not continue long enough for logistic issues to affect CIC teams' decisions. However, ENWGS logistics models would have surfaced these concerns eventually.

IMPRESSIONS AND LESSONS LEARNED

Our impressions and the lessons learned from this integration effort are as follows:

BG Training - The WHO

BG training needs to be flexible to support any potential participant. For instance, although a BG commander may not wish to participate in several ships' prosecution of a hostile submarine, his decisions should be included either by an exercise coordinator or by automation (surrogate participant). Similarly, a member of the CWC team, such as the anti-submarine warfare commander (ASWC), may not want to participate in an "anti-air only" engagement exercise, but his demands for carrier deck space for ASW aircraft should impact the anti-air warfare commander. Numerous training systems, including ENWGS, make allowances for missing or surrogate participants using advanced expert system-like technologies.

BG Training - The WHAT

BG training systems must support flexibility — multi-platform, multi-warfare, and multi-dimensional (time) — to accommodate all of the characteristics of a BG. Training systems should not limit the design of BG exercises. Exercise design should be at the discretion of the exercise sponsor.

BG Training - The HOW

Admiral Mustin stated, "We ought to train as we expect to fight." The only way to do this is to simulate and stimulate all of the BG information flows to the WHO (e.g., fleet commanders and staff) in the WHAT (e.g., aircraft carrier, support ships).

BG Training Systems

The simulation system should be flexible enough to support either individual (standalone embedded) ship training or the ship's integration into multi platform BG exercises. If a ship chooses to conduct standalone training, the training needs to be accomplished in the context of an entire BG environment.

Shipboard Combat Systems

Although the basic reference NTDS documentation, OPS-411, has training messages to allow for simulated contacts over many different platforms, each ship's combat system with which we interfaced accepted different message types. This was learned at testing. Further, none of the three ships' combat systems with which we interfaced could accept subsurface training messages. Thus, we could not display subsurface contacts on the ships' consoles. Subsurface contacts were portrayed via ENWGS ASTABs.

Simulation/NTDS Transmission

Our initial estimation that a 9600-baud telephone line with STU III encryption would be adequate to distribute both the simulation and NTDS data link proved to be valid. The single point of entry into the ship's combat system was the NTDS data terminal set (DTS). The procedure of connecting/disconnecting NTDS DTS cables proved to be tedious and potentially upsetting to the ship's NTDS program. A more integrated method is required.

Ownship Activity

Our estimation that 128 active contacts per ship would be adequate to fully train a ship's combat crew was generally accepted by the end users. Many passive ESM and sonar sensor contacts accompanied the 128 active sensor contacts.

FUTURE RESEARCH

This project provided us with an understanding of the BG training environment and the available technology to enhance it. This understanding has revealed several areas ripe for future research in the BG training continuum.

Increased Simulated Platform Performance

There is little doubt of the need to enhance exercises within simulated platforms. Simulated platforms need to function as close to the real world as possible. In our approach, simulated, friendly NTDS units (such as the E-2C) transmitted only active sensor contacts, their full and automated participation in the exercise, including automatic reaction to NTDS force orders, still needs to be demonstrated. The basic research question to be addressed is, "How can real-world activity be fed back to the simulated world?" Additionally, research conducted by the training community will enhance the "behavioral" characteristics of simulated platforms and surrogate players to ensure realism in the training environment.

Human Aspects

Although BG exercises are currently being designed, conducted, and debriefed in both shore and at-sea environments, the fully integrated simulation/live BG exercise of the future calls for a new generation of exercise design, coordination, and monitoring features, including design aids and automated scenario generation. This will decrease the training overhead costs of support personnel and increase the training benefits.

Simulation Support At-Sea

There is a desperate need to increase the realism in BG training at-sea. While there is no substitute for actual flying or steaming hours, shipboard combat team training could be significantly enhanced by integrating simulation and real world scenarios. One live B-52 bomber launching a mock, hostile cruise missile attack on a BG could be "enhanced" to appear to the shipboard console operator, and thus the entire CIC team, as an entire Backfire regiment. Even in the "New World Order," the need to train these operators in BG operations will remain.

We should consider air and submarine platforms, as well as shipboard C^4I and other simulation devices, which add value to the long term BG training goal, when implementing technology enhancements. Value added technology increases training realism and exercise support, such as game design aids or post exercise analysis. This would include new shipboard embedded part task simulators such as the SQQ-89 Onboard Trainer (OBT). (Refer to Figure 1.) The technology available in the tactical air combat training system (TACTS) and the

Global Positioning System (GPS) also add value. By careful integration of existing C^4I and simulation technology, we can create the realism we need to train as we expect to fight.

Joint Warfare

There is a growing interest in further expanding the BG training continuum into the joint warfare environment. General Kelly was recently quoted as saying, "ENWGS was used to provide naval play in a pre-Desert Shield exercise, which previewed operations of U.S. forces in the Gulf area and effectively prepared commanders [General Schwarzkopf and his Central Command Staff] for those operations." The exercise that General Kelly referred to was previewed several months prior to actual deployment and without the benefit of automated interfaces to the other simulation systems that participated. What further benefit could have been gained by General Schwarzkopf if he had conducted these Gulf-area exercises and wargames immediately preceding open hostilities? These benefits might be realized as each service's "islands of simulation" are integrated. Joint warfare simulation integration is one of the charters of the Joint Warfare Center (JWC).

The Ultimate Design

The ultimate BG training system should provide maximum flexibility to automatically design and conduct BG exercises based upon objective BG training requirements. Variables to be considered in this design are: WHAT - which platforms, real and simulated, to include; WHO - which people need to be trained; and HOW - which systems need to be included. Exercise designers and those who sponsor BG exercises should expect no less. All things to all people - why not? Clearly, the state-of-the-art in systems integration is at hand to provide this.

ABOUT THE AUTHOR

Larry Morton is a graduate of the U.S. Naval Academy, holds an M.S. in Information Systems Management from USC and is a Ph.D. candidate in Information Studies at Drexel University. He is currently a Senior Computer Scientist at Computer Sciences Corporation and was the leader of this project. Prior to joining CSC, Mr. Morton was the wargaming analyst at Commander-in-Chief, U.S. Pacific Fleet Headquarters. While on active duty, he was a qualified tactical action officer on NTDS and non-NTDS ships and served as his ship's CIC and training officer. He is currently a Commander in the Naval Reserves.

EMBEDDED TRAINING
for
ARMORED SYSTEMS MODERNIZATION

Lester D. Curless
Engineering Branch Chief

Traci A. Jones
Systems Engineer

Richard A. Copeland, Jr.
Systems Engineer

Department of the Army

Project Manager for Training Devices
Research & Engineering Management Division
Orlando, FL

ABSTRACT

In an effort to reduce training costs for sustaining soldier proficiency in deployed units, the U.S. Army Training and Doctrine Command (TRADOC) has identified embedded training as the preferred alternative to be considered for development of training systems used to prepare and sustain future armored vehicle crew members. Prior to full scale development, the demonstration/validation portion of the vehicle acquisition process must investigate the optimum implementation of embedded training for the next generation of armored combat vehicles. This paper reviews the general goals, and some of the challenges involved, for embedded training within the six future vehicle systems planned for the Armored Systems Modernization program; the paper focuses primarily upon the present efforts directed for developing embedded gunnery and tactical simulation into the electronics of two of these vehicles, the Block III tank and the Advanced Field Artillery System (AFAS).

Background

In view of the constantly uncertain and turbulent situations in Europe and Third World Nations, the U.S. Army has initiated the Armored Systems Modernization (ASM) program to ensure the fielding of a competent, combat effective conventional land force able to engage the threats anticipated in the 21st century. This ASM program is intended to take advantage of emerging technological opportunities which will be applied to emphasize commonality of vehicle needs and sustainment costs for the group of vehicle systems.

The ASM program includes the six new vehicles, shown in Fig 1, which are being developed to replace existing armored systems.

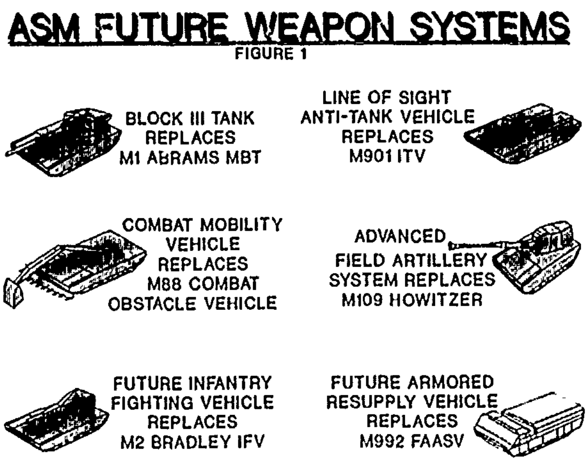


FIGURE 1

Integrated Training Systems Approach

According to the TRADOC vehicle proponents, each ASM vehicle variant requires an Integrated Training system (ITS) consisting of varied combinations of the following kinds of training devices, simulators and simulation (DSS) (see Fig 2):

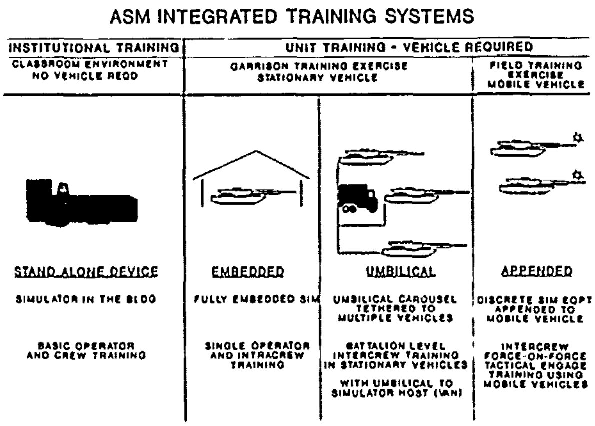


FIGURE 2

Embedded Training System (ETS): The TRADOC definition of embedded training is "Training that is delivered by capabilities built into an operation system in addition to the primary function. The training is made available by components of the equipment that take advantage of the overall system capabilities. It can train individual, operator, crew, functional and force level tasks."

The ETS is envisioned to be tailored to the needs of the respective ASM vehicle crew operation and maintenance tasks. Refer to Fig 3 for a simplistic block diagram of ETS operation.

DETACHED VEHICLE ET MODE
SYSTEM BLOCK DIAGRAM

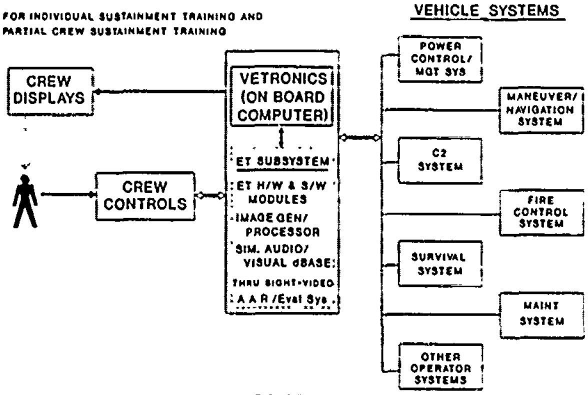


FIGURE 3

Umbilical Carousel Trainer (UCT): External simulation hardware and software equipment that can be connected via an "umbilical" cord to the vehicle ETS. The UCT would have the additional capacities (beyond those of the ETS) to provide some of the desired training features and mission complexities that are not deemed cost effective for embedding in the vehicle. The UCT is conceived as an external simulator device that would provide any individual vehicle crew the capability to collectively train with other individual vehicle crews in battalion-and-below (B2) level of command and control exercises. This UCT would be configured to allow any vehicle of any of the six ASM variants to connect to the UCT, thus providing an opportunity to train complete crews of any ASM vehicle together in a collective team training scenario. UCT could be a mobile, van-type of simulator, complete with separate instructor(s) stations and detailed training data base to address various levels of operator proficiency, (i.e. basic, transition and sustainment). Each UCT is envisioned to be posted to a battalion for use within the garrison (e.g. in the motor-pool area) for ease in coordination of the usage by different types of vehicles. The UCT would accommodate a minimum of 12 vehicles at a time, in a "carousel" fashion. The UCT would have the capability of accessing Simulation Network/Close Combat Tactical Trainer (SIMNET/CCTT) via direct or radio frequency connection.

Appended Training System (ATS): External simulation hardware and software that is attached externally and plugged into the vehicle electronics, or actuated externally, during mobile field training exercises for realistic tactical engagement simulation which includes weapons effectiveness along with the aural/visual ("flash and boom") impact cues of battle explosions. Present-day examples of ATS include the Main Tank Gun/Weapons Effect Signature Simulator (MTG/WESS) and Simulation of Area Weapons Effects (SAWE) type of "non-system" simulation equipment. Currently appended tactical engagement simulations like Multiple Integrated Laser Engagement System (MILES), Tank Weapon Gunnery Simulation System (TWGSS) and Mobile Independent Target System (MITS) are anticipated to be designed into the ETS of the vehicle.

The "external" nature of ATS is conceived to allow actual movement by the "own" vehicle during field exercises at Combat Training Centers, whereas the "external" nature of UCT is to afford the opportunity for multiple stationary vehicles to interconnect for collective team or "task force" training in garrison environments.

Institutional Gunnery Trainer (IGT): Equivalent to current stand alone devices which provide Conduct Of Fire Training (COFT) to the gunner/commander crew operators of tanks and infantry fighting vehicles. IGT is used at the respective vehicle proponent schools primarily to provide basic and transition gunnery training for crew operators. IGT is expected to provide more opportunity than presently available in COFT to train the vehicle driver for vehicle maneuvering during gunnery exercises. Refer to Fig 4 for institutional simulator block diagram.

STAND ALONE DEVICE TRAINING MODE
SYSTEM BLOCK DIAGRAM

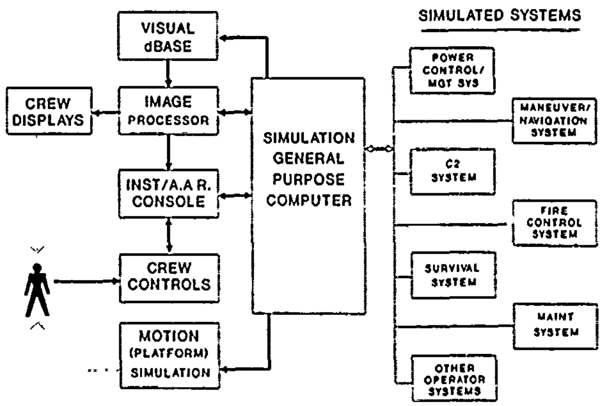


FIGURE 4

Institutional Driver Trainer (IDT): Similar to IGT, except that the proponent school is providing basic and transitional training to the vehicle crew driver. IDT should take advantage of the common driver controls and displays expected to be duplicated across the common chassis design of the ASM vehicles.

Institutional Maintenance Trainer (IMT): Provides an assortment of troubleshooting panels to train maintenance procedures for ASM vehicle engines, transmissions, hull electrical,

fire control systems, chassis hydraulics and turret controls. The IMT is envisioned to emphasize the common maintenance design of the ASM common chassis and Vetronics Built-In-Test (BIT) circuitry. The IMT should provide training at the unit and intermediate levels of responsibility.

Close Combat (Crew) Tactical Trainer (CCTT): Provides institutional training, at the basic and transitional level of proficiency, in the combined arms tactical procedures (emphasizing command and control) to be implemented in the future Air Land Battles. The CCTT could serve as the proponent school's version of the UCT collective team training in the garrisoned unit. A mobile configuration of CCTT should be available to meet proponent institutional requirements for "transportable" crew trainers.

Each ITS is required to be Ready for Training (RFT) at various CONUS/OCONUS locations at least one quarter prior to the vehicle Initial Operating Capability (IOC). The ITS requires an Instructional Strategy and Systems Engineering design for all embedded training functions, training related (stand alone) DSS, as well as the Programs of Instruction (POI) and related courses, courseware and courseware development systems (Authoring).

In accordance with the ASM system Operational & Organizational (O&O) Plan, the preferred training capability will be embedded to the maximum practical extent.

A fully tested, validated and verified ETS is required to provide sustainment training and familiarization within the vehicle for crew members assigned to operate and maintain their respective ASM variant vehicle. Let's now focus on some of the specific interface challenges which accompany the desired ETS performance.

Standard Vehicle Electronics Architecture

ASM vehicle technology will emphasize a modular, digital electronic Vehicle Control and Operation System (VCOS) whose configuration has been designed to use the Standard Army Vetronic Architecture (SAVA), presently being developed by the Army Tank Automotive Command (TACOM). SAVA is based upon an assortment of modular electronic processor boards which exchange data across six different buses employing standard interfaces. It is intended that these modular boards and buses will maximize design commonality across the six different vehicles, thus reducing development and sustainment costs for the modernization of this group of vehicles.

The ETS is based on emulating, or alternatively stimulating, the same VCOS electronic signals and data that would normally be cued to each of the vehicle crew members in the course of their assigned duties within the vehicle. The ETS will take advantage of any bus data and interface that is normally processed in the VCOS during combat (non-training) mode. The ETS will stimulate those available VCOS functions and simulate those that are not normally available in the combat mode, but are required during the non combat (training) mode of operation for the vehicle.

The ETS must, therefore, have access to all of the analog and digital signals being communicated on all of the SAVA vehicle buses, including external to Lowest Replaceable Unit (LRU) type:

a. Utility Data (Low Speed) Bus: Power management and control, simple sensor, gauge and actuator monitor and control. Less than 1 MHz.

b. Video Data Bus: Used for imaging and nonimaging sensor video at the crew stations displays.

c. High Speed Data Bus (HSBD): Used for the most complex and high performance vehicle applications having a flow rate exceeding ten megabits throughout (20Mhz).

d. MIL-STD-1553B Medium Speed Data Bus (MSDB): Used for the moderate complexity data performance capability less than ten megabits throughput. (Less than 1 MHz).

Internal to LRU type signals:

e. VETRONICS Test and Maintenance (T&M) Bus: Used for access to, and simulation of, the diagnostic data required to troubleshoot vehicle systems.

f. VME (SAVA) System Backplane (SSB) Bus: for inter-processor/peripheral communications, userdefinable input/output channels and interconnect to all external data buses.

Fig 5 provides a basic diagram of ETS access and function on the SAVA/VETRONICS bus.

ETS ACCESS TO SAVA/VETRONICS BUSES

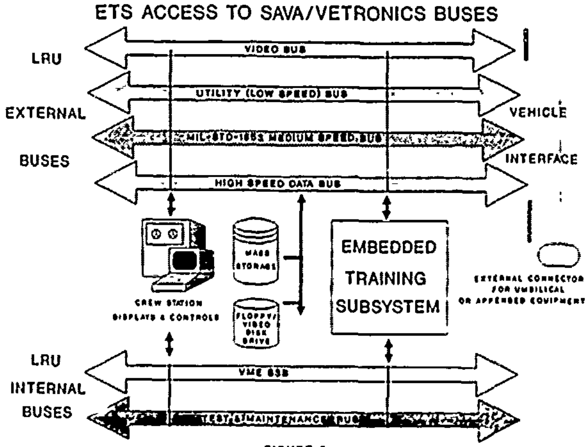


FIGURE 5

Overall Vehicle Mission Performance

The ETS would simulate accepted versions of operational missions, in both normal and degraded modes of vehicle operation. Operational missions would vary for each ASM variant. The operational scenarios for each variant should provide necessary sustainment level of proficiency to individual crew members, both operators and maintainers. Proficiency must include both individual part-task type of training as well as collective team or

force-on-force tactical training. The ETS must include the capability of free-running mission scenarios with which the crew member would dynamically respond as dictated by normal combat procedures and doctrine. These scenarios would conclude with an After Action Review (AAR) type of feedback, which provides the student with appropriate guidance and evaluation regarding performance in response to the complete scenario.

The ETS must be designed to endure the same physical environment as the remainder of the vehicle electronics (Vetronics). The ETS must also replicate the threats against which the ASM vehicle must respond. Like the remainder of the VCOS Vetronics, the threats to the ETS are dependent on the mission requirements for the particular type of vehicle system in which it is installed. The most likely and most severe threats are those which ETS would encounter in its employment in tank armored vehicles. The design of ETS shall therefore take into account the requirement that the crew, vetronics subsystems and VCOS subsystems, including ETS, respond effectively to the following enemy threats:

a. Weapons - including anti-vehicle guided weapons, land mines, artillery, tanks, attack aircraft, laser weaponry, and small arms fire.

b. Anti-vehicle obstacles.

c. Electronic Warfare (EW) - including disruption (jamming), deception, destruction of communication nodes, and exploitation of communication information (regarding force strength, locations and intention).

d. Nuclear, Biological and Chemical (NBC) Warfare - including nuclear electromagnetic pulse (EMP).

e. Directed Energy Threat technologies such as laser and High Power Microwave (HPM).

ASM vehicles are expected to use high-mobility tactics and closely-coordinated assaults to improve their combat effectiveness against enemy targets in this threat environment. The enemy arsenal is expected to be used to its fullest to counter the vehicular movement, to disrupt communications used for coordination, and to destroy communication nodes. Thus, radios can expect heavy jamming, and radio transmitters can expect to draw incoming missiles or projectiles. Those emissions that are not jammed and do not draw fire will be the ones from which the enemy can extract tactical information. Optical devices and human eyes will be targets for enemy lasers. Therefore, to minimize the threats, the vehicles must use secure, anti-jam communications which minimize compromising emanations while exposing a low electronic profile. In addition, EMP effects must be controlled and contamination from NBC attack neutralized or accommodated. The ETS must not only survive such threats, but it must also be able to replicate these threats for training and crew preparation purposes.

The vehicle system's ETS shall include an Embedded Technical Help (ETH) capability. ETH will provide operational checklists, automated logbook, parts requisition capabilities, built-in-test, preventive maintenance procedures, automated field manual reference guides, and on-board resource inventories. ETH is conceived to be an automation system which provides labor or material saving procedures that aid crew member performance on the job. While ETH may also serve as a training aid, it is not conceived to be a system of providing sustainment training to ASM vehicle crew operators and maintainers. ETS shall include the capability of training crew personnel in the use of ETH as a tool in their normal duties.

User Desires for ETS Performance

User requirements for the ETS suggest that it will be used both in protected garrison locations and the unprotected environments of actual combat assembly areas, as well as in the harsh environments associated with force-on-force (FOF) training at Combat Training Centers. The vehicle combat laser detector and transmitter may therefore be concurrently used for tactical engagement simulation during FOF training exercises. This embedded feature would then replace the appended nature of present MILES or TWGSS type of laser-based training equipment.

The ETS would simulate realistic combat scenarios which provide proper visual and aural stimulus of, and response to, the complete vehicle crew actions in a manner duplicating the performance expected during actual combat operations. The ETS shall be designed as part of the vetronics and shall be interfaced directly to the crew member controls and displays in order to provide an assortment of readily available combat training scenarios that directly simulate all of the vetronics functions.

The ETS operation must be transparent to the vehicle crew and should not interfere with the crew's normal operational functioning of the vehicle. The ETS design must ensure no inadvertent access to the normal combat mode of fire control operation while conducting training; nor should there be any inadvertent access to the training mode of vehicle operation while conducting actual combat operations. The vehicle operator(s) should be able to access the ETS from the combat (i.e. non-training) mode of operation in less than five minutes.

The ETS design must maximize the use of normally available vetronics resources such as power supply, crew controls and displays, memory, audio and visual generators/processors, disk drives etc. in order to minimize electronics hardware space claim, with subsequent additional cost, required by the ETS. Maximized use of the normally available resources is also required in order to ensure direct emulation of the typical visual and aural stimuli expected to be provided to the operator during actual combat mode demonstration.

ETS Functions

The ETS would include, as a minimum, the following performance modules or components:

a. Training Menu/Mission Selection Module (TMSM)

b. Crew Station Display Module (CDM)

c Crew Station Operator Controls Module (COCM)

d. Audio Control Module (ACM)

e. Centralized Data Processor Module (CDPM)

f. Decision/Response Branching Logic Module (DBLM)

g. Performance Monitor/Evaluation Module (PEM)

h. Appended/Umbilical Interface Module (AUIM)

The term "module" is meant to denote a hardware and software unit which is dedicated in design and function to perform the described capability. It may include one or more electronic circuit boards; it may be contained as a part of a single circuit board which is populated by integrated circuits performing various other "module" functions. The software involved may be, for example, a subroutine of higher ordered modules.

Training Menu/Mission Selection Module (TMSM): This module would enable the student or instructor to select and setup the type of training (individual, crew, tactical, level of help, etc.) and particular mission scenario parameters (threats, weather, degraded equipment, etc.) to be simulated. This menu/selection information would be sent via the CDM to the appropriate operator's display(s). Module input comes from the CDM, COCM and CDPM.

Crew Station Display Module (CDM): This module would receive and format the appropriate visual information (video image or graphical symbols or text) to be cued to the respective crew station operator. This information would be output via the CDPM to the high speed digital data bus for viewing at the operator's display(s). Module inputs come from the COCM, TMSM, PEM, DBLM and CDPM.

Crew Station Operator Controls Module (COCM): This module would receive, interpret and translate the digitized data being sent from the operator's controls. This data would provide the input to simulation algorithms that provide movement or action cues to the operator which are commensurate with the perceived operator control. The output of this module would usually be sent to the TMSM, CDM and DBLM for providing the cued visual perception of the operator response. Module inputs come from the CDPM.

Audio Control Module (ACM): This module would enable/disable normal crew audio communication. It would include a synthesized speech voice which could serve for either "missing" crew members, personnel external to the vehicle (e.g. battalion radio operators) or an instructor. Output from this module would usually be sent to crew member headsets via the CDPM, audio couplers and the MDSB. Module inputs come from the CDPM and PEM.

Centralized Data Processor Module (CDPM): As the name suggests, this module would be central monitor ("bus watcher"), interpreter, access path and supervisor of ETS digital data being input or output. This module must access all digital buses, both external and internal. Module outputs go to all ETS modules and to all vetronics buses; module inputs come from all buses and from the TMSM, CDM, ACM, and AUIM.

Decision/Response Branching Logic Module (DBLM): This module would interpret the action taken by the operator versus a series of branching logic trees to provide the appropriate response to the operator. Input comes from the CDPM (decision baseline for comparison purposes) and COCM; outputs go both to the CDM, as well as to the Performance Monitor/Evaluation Module.

Performance Monitor/Evaluation Module (PEM): This module includes an artificial instructor capability that would record the operator's responses to various threats. The record could, as chosen by the operator during the training menu phase, be either:

a. a complete audio and visual record of the free-running scenario just undertaken with summarized audio-visual critique provided at the end of the mission, or

b. step-by-step intervention by the instructor to evaluate or guide each action taken by the operator in response to each obstacle or threat.

To the extent possible, artificial instructor capability must use simple graphics and other easily comprehended formats for presenting feedback to the student(s). This module would also serve as the Integrated Training Management System which provides automated selection of training exercises/scenarios on the basis of predefined instructional strategies and operator/crew past performance. Module inputs come from the CDPM (mission completed, going directly to AAR) and DBLM; outputs go strictly to the CDPM for storage of data record or for the appropriate displayed response to the operator via the CDM/ACM.

Appended/Umbilical Interface Module (AUIM): As the name suggests, this module serves as the central protocol handler of data coming from, and going to, the training simulation equipment which may be either appended to, or actuated by, the chassis during actual moving vehicle, Operational TEMPO (OPTEMPO), exercises or connected via umbilical cabling during stationary "motor-pool" training environments. The module processes both inputs and outputs going between the external training equipment and the CDPM. It is expected that this module would access the external training equipment via an external chassis "umbilical" connector(s).

Physical Characteristics

In order to present the least burden on already taxed vehicle resources, a primary goal of the ETS shall be to minimize weight, volume and power consumption. The ETS design should allow for a modular capability thereby allowing flexibility to install ETS in any selected vehicle, as training is to be conducted.

The ETS design will comply with the SAVA circuit board formfactor construction based on ANSI/IEEE Std 1014-1987 Double Eurocard with VME Backplane (approx 9.5" x 6.3"). Each of these Double Eurocard/VME Backplane circuit cards would utilize through-the-hole Printed Wiring Board (PWB) construction, consume no more than 15 watts of electrical power (dissipation by convection cooling) and should weigh no more than 2 pounds.

The ETS requires a computer generated image generator/processor, which will be compatible with the crew station(s) display technology. The ETS is perceived to require both a mass storage device and a floppy diskette drive device for loading of the various mission scenarios and part-task instruction courseware. The image generation equipment and storage devices may be jointly required for other VCOS functions and not a unique asset solely for the ETS.

Extended Range Gunnery Fire Control Demo

The Armament Research and Development Engineering Center (ARDEC) has initiated the Extended Range Gunnery Fire Control Demonstration System (ERGFCDS) in an effort to examine the potential technological performance expected for the Block III tank fire control.

In this regard, the ERGFCDS contractor (Texas Instruments) will analyze the impact of incorporating the simulation required for embedded gunnery training into the fire control processor functions.

If the analysis suggests acceptable impact, then Texas Instruments shall develop and demonstrate an ETS that is based on simulation hardware and software which is built into the actual fire control electronics. This ETS demonstration is intended not only to assess risk for full scale development of ET in the ASM vehicle, but also to provide a useful tool to assist Government personnel to learn the operational procedures for the proper use of the various vehicle subsystems during eventual test and integration efforts.

Visual Subsystem. This visual subsystem for ET will simulate tactical scenes consisting of European summer (as well as desert) terrain, man-made cultural features and the full gamut of vehicle critical crew operations; (i.e. target engagement for the tank, obstacle elimination for the CMV, etc). Scene content would be variably occulted to realistically simulate day, night and limited visibility (smoke, dust and haze) conditions. The simulated terrain should provide target presentations in various degrees of exposure (full, partial, intermittent or hidden) and aspect angle relative to the view of the own vehicle.

The visual subsystem design should be based on an optimum consideration of state-of-the-art technologies including, but not limited to, CGI (Computer Generated Imagery), CD-ROM (Compact Disk-Read Only Memory), optical disk, and virtual reality. Simulation visual imagery coloration should match that of actual vetronics visuals viewed under combat conditions.

Instructional Subsystem (IS). IS for ET will be menu-driven to provide for the easy selection of desired training exercises in which the crew members interact with each other to perform combat operational procedures. The IS should monitor and evaluate the performance of the student(s), providing a report to the student of past performance, accompanied with complete replay capability of the preceding exercise. Replay of the training exercise may be selectably paused for instructor comment. The performance report should include a comparison of the "acceptable" performance standard for each element versus the actual performance achieved. The performance report will provide constructive criticism intended to remediate student performance. The design of the IS should include artificial intelligence technology.

The IS software is expected to gene. .e menus allowing the crew members to select scenario(s) which provide training at different levels of progressive difficulties. IS would include performance standards and scoring criteria for pass/fail/remedial comment of student performance.

ETS design should demonstrate the capability for modification/updating of training exercises (tutorials and scenarios) by Government personnel in order to ensure fielding of training changes to deployed vehicle units in less than two months.

Scenario content. Scenario content sequence is expected to provide randomization of target(s) or obstacle(s) location(s), target or "own" vehicle routes, mobility status and order of occurrence to prevent the negative training involved with student anticipation of the sequence of previously encountered scenarios. For example, if a student repeats the same exercise with which to train, he should be confronted each time with a different sequence of targets (or obstacles) to be detected, acquired and engaged. Randomization of scenario content should not preclude identification of critical scenario parameters during the AAR for each exercise repetition.

For those vehicles primarily involved with engagement of targets, the ETS would generate at least one simulated combat scenario for each of the following situations:

a. Stationary own vehicle, stationary targets.

b. Stationary own vehicle, stationary and maneuvering targets.

c. Maneuvering own vehicle, stationary targets.

d. Maneuvering own vehicle, stationary and maneuvering targets.

Each scenario containing maneuvering targets should include at least one maneuvering rotary-wing attack aircraft. The simulation software would provide the student with the proper cue of a "killed" target. Likewise, the student shall receive proper cues of hostile targets firing on and "killing" the student, based on improper target engagement by the student. Once the scenario is begun by the student, the crew controls and displays should perform in the same manner of fidelity as that performed during normal combat (i.e. non-training) mode. Selected scenarios might include a wingman vehicle that is visibly maneuvering in the gunnery/tactical exercise. This automated, simulated wingman vehicle will provide the student(s) with identification training of friendly versus hostile targets. An additional benefit will be to provide tactical training for the own vehicle to interact with a simulated mobile wingman which is also engaged with targets.

The ETS scenarios would be based upon simulation of actual combat missions which are

expected to be performed by the ASM vehicle. Prior to design of the simulation scenario, Texas Instruments will provide a narrative "storyboard," outlining each proposed mission to be simulated. The storyboard would describe the appropriate crew procedures to be employed, the mix of targets/obstacles to be encountered.

ETS is expected to include the provision of narrative tutorial information to allow the student to selectively review basic crew control functions and operational procedures. This tutorial is intended to get the student prepared on vehicle system operation prior to beginning a simulated combat scenario. This tutorial should also address any diagnostic or maintenance procedures tasked to each crew member (e.g. use of self-test or Built-In-Test-Equipment procedures).

Each scenario would include a Situation Report (SITREP) on the own vehicle status and starting conditions at the beginning of the scenario. The SITREP will include, but not be limited to:

a. Indication of fully operational or degraded equipment on own vehicle.

b. Type and quantity of ammunition available and presently indexed in auto loader.

c. Mobility and defilage/enfilade status.

d. Present selection/state of crew control switches for each crew member, including menus/information being displayed.

e. Present manning of crew positions and designated sectors of responsibility.

f. Intelligence report of possible threats and expected tactics.

g. Visibility and time-of-day conditions.

h. Outline of orders received from echelon commander.

i. Fuel availability.

The SITREP will be preceded by a training exercise outline which describes the type of forthcoming scenario and training exercise to be completed, general description of expected individual and collective crew performance and the standards expected to be attained (e.g. "acceptable" performance). Each outline must provide crew instructions which give the purpose of the respective exercise and discuss specific gunnery and tactical skills to be trained by the respective exercise. The outline is intended to be general in nature and must not allow the student to predict exact location/sequence/mobility of targets to be presented within the subsequent scenario.

Simulated target hits and kills will be based upon the most current probabilities of hit and kill data available for the types of munitions and targets employed in the scenarios.

Scenario content would include exercises which provide specific training in the use of the vehicle Battalion and Below Command and Control (B2C2) system. Proper B2C2 procedures should be incorporated into the scenario(s) to provide comprehensive training of realistic combat requirements for integrated action by the crew member(s) to use B2C2 tactics during target/obstacle engagements.

Advanced Field Artillery System

The Advanced Field Artillery System is one of the six Common Chassis ASM variants. AFAS, with many automated features, will replace the M109 family of howitzers. The Program Manager for the Advanced Field Artillery System (PM AFAS) has initiated an AFAS Advanced Technology Transition Demonstrator (ATTD) contract.

The ATTD phase of the AFAS program precedes the fabrication of a prototype vehicle. The current AFAS ATTD requires definition of the embedded training concept for maintenance and operator tasks. The identification of subsequent training tasks that can be demonstrated in the ATTD or prototype phase is also required.

The AFAS is ideally suited for embedded training. The AFAS, as an indirect fire weapon, will not require the complex embedded visual subsystem anticipated on the direct fire ASM variants, such as the Block III tank. The AFAS crewstations shall include redundant displays and controls. If the analysis by the ATTD contractor warrants, the use of one or more of the onboard crewstations as an instructor station could be demonstrated in the ATTD or prototype. An AFAS crewstation reconfigured as an instructor station, via a training software load or some other means, could control a "closed loop" training simulation. The ability to perform such a "closed loop" training simulation without the need for appended or umbilical devices would demonstrate the utility of fully embedding certain portions of the total training system. The ability to network via an umbilical device, with one AFAS manned by instructors and other AFAS(s) manned by students, could provide platoon and battery level training exercises.

Advanced Technology Transition Demonstrators

Prior to construction of any prototype vehicles, Advanced Technology Transition Demonstrators (ATTD) will be developed to serve as vehicle "test beds" to evaluate the varied technological features being considered for each ASM vehicle.

Each ATTD (Fig 6) will include a crew station "mission module" attached to an actual vehicle functional chassis, which has been designed to incorporate maximum commonality of armored vehicle chassis components (i.e. propulsion system, modular armor, etc.). The ATTD will thus be a mobile evaluation tool, allowing target audience crews to sample the features proposed to be incorporated into the full scale development vehicle.

The ATTD will be preceded by a System Integration Laboratory (SIL) effort which serves as the "hot bench" mockup of the features to be installed within the mission modules.

The Component Advanced Technology Test Bed (CATTB) will serve as the mobile test bed used to evaluate certain technological advances expected for future tanks, such as the ERGFCDS features, including embedded training. CATTB will use an M1

ADVANCED TECHNOLOGY TRANSITION DEMONSTRATOR

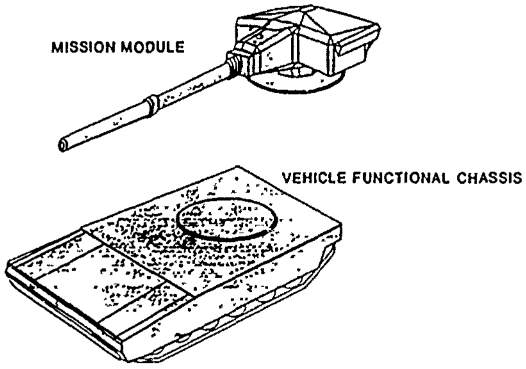


FIGURE 6

Abrams tank chassis which has been modified to reflect SAVA electronic modularity.

Concept Formulation Process

The Project Manager for Training Devices (PM TRADE) is the U.S. Army Material Command agency tasked to investigate the practicality of developing embedded training for ASM vehicles. PM TRADE analysts, logisticians and engineers are reviewing the user requirements for ET, in view of the technological advances expected at the time of vehicle development and in view of feasible approaches which may serve as alternatives to the depth of training desired to be incorporated into the ETS. Fig 7 depicts the "cauldron" of trade-off analysis which is ongoing to consider all of the known information pertinent to embedded training considerations from other DOD programs.

AREAS TO BE INVESTIGATED
for ASM EMBEDDED TRAINING APPLICATIONS

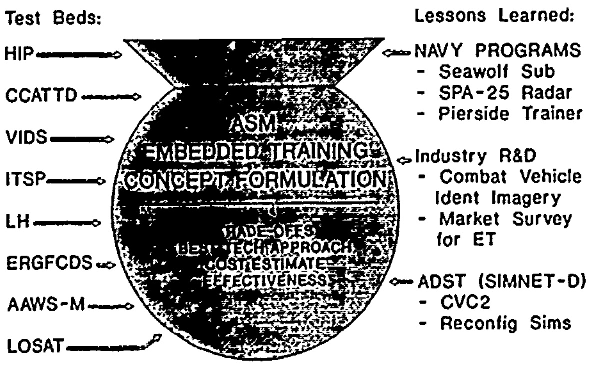


FIGURE 7

PM TRADE's Concept Formulation Process (CFP) will determine and analyze the various alternatives which are deemed feasible for ET and will examine ET as a potential solution to the total ASM integrated training requirement. The resultant analysis to be conducted by TRADOC with PM TRADE will yield answers to such questions as:

a. What are the critical operator tasks and standards to be trained using ET?

b. What impact will ET have on the Reliability, Availability and Maintainability requirements of the crew station controls/displays due to the anticipated increased usage?

c. What amount and type of training can acceptably be stored within the vehicle resources?

d. Should the ET scenarios and database be stored in "soft" drive format (e.g. floppy disc) and loaded into the vehicle computer only when training is about to be conducted?

e. What will vetronic technology provide during the next five years to allow the increased data flow rate and capacities required for the interactive intervehicle crew training desired for user collective training?

f. Should the vehicle laser range finder be jointly used as a range finder and also as an embedded MILES/TWGSS-like transmitter? Should radio frequency techniques replace the present laser-based tactical engagement simulation?

g. Is ET less expensive than our present reliance on stand-alone training devices? How cost effective is ET?

Answers to these and other questions will be pursued during the Concept Formulation Process to determine the wisest assessment of ET for each ASM vehicle system. The advantages and disadvantages displayed in Fig 8 must be evaluated to determine the optimum ET requirements for each ASM variant.

TO EMBED ... OR ... NOT TO EMBED

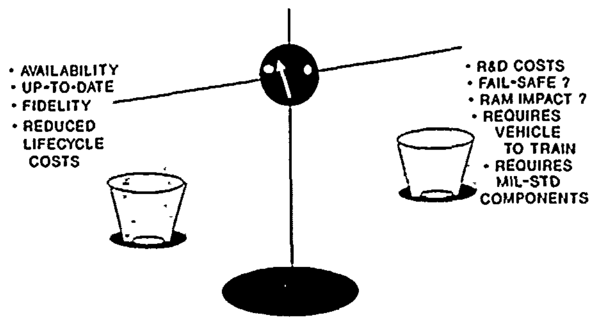


FIGURE 8

Acknowledgments

The bulk of this paper has been extracted from PM TRADE ET specification PMT-91-S001, prepared by the authors; reference to the draft SAVA system specification MIL-V-62626 (24 April 1989) has been made for SAVA design details. Specification review and comments coordinated by the Combined Arms Training Activity (ATZL- CTT) and by the Block III Program Office (SFAE-ASM-BT) are appreciated. ASM program information regarding background, schedules and vehicle configurations have been provided by the office of DPEO-ASM-F and by the Combined Arms Combat Development Activity.

About the Authors

Mr. Lester D. Curless is presently the Branch Chief for the Engineering Management of Network Simulators at PM TRADE; he was the former lead PM TRADE systems engineer assigned to the ASM Program. During the last six years, he has served as project engineer, lead engineer and project director of various simulator programs for both the U.S. Army and Navy departments. Prior to his involvement in simulation and training devices, his career centered on the supervision, over a ten year period, of the electrical and electronic engineering efforts required for the modification, repair and testing of various military aircraft. Mr. Curless is a registered professional engineer and holds a Bachelor's Degree in Electrical Engineering awarded by the University of Florida.

Mrs. Traci A. Jones is presently assigned to PM TRADE as a systems engineer for the ASM program. Past assignments include project engineer responsibilities for several U.S. Navy surface ship training systems. Prior to her involvement in simulation and training devices, she served as a systems engineer and program manager for data acquisition in support of U.S. Navy Submarine Fleet Ballistic Missile testing at Cape Canaveral, Florida. Mrs. Jones is a certified Engineer-In-Training and holds a Bachelor's Degree in Electrical Engineering awarded by the University of Central Florida.

Mr. Richard A. Copeland, Jr. is an ASM systems engineer in the Engineering Management Branch for Battlefield Operation Systems of PM TRADE. Prior to PM TRADE, he spent six years at the Naval Training Systems Center, most recently as the S-3B Viking Tactical Systems Project Engineer. Mr. Copeland is a certified Engineer-In-Training, holds a Bachelor's Degree in Electrical Engineering from the University of Central Florida, and is working on a M.S. degree in Simulation Systems. He is a member of Tau Beta Pi and Eta Kappa Nu.

ELECTRONIC WARFARE CONTINUUM ASSESSMENT PROGRAM FOR NAVAL AVIATION

Maureen L. Bergondy and Katrina E. Ricci
Naval Training Systems Center, Orlando, FL

ABSTRACT

The need to document warfighting readiness and training effectiveness is a major concern for warfare sponsors, operational commands and training system developers. The Electronic Warfare Continuum Assessment Program (EWCAP) is a low cost method for rapid evaluation of electronic warfare (EW) readiness and training effectiveness across the careers of Naval Aviation personnel. EWCAP provides documentation of EW performance and training deficiencies, and recommends solutions to identified training deficiencies. To produce a snapshot view of EW knowledge and skills, microcomputer-based tests have been developed and administered to the EA-6B, E-2C, F/A-18, and A-7 communities, and are in development for the S-3, A-6 and F-14 communities. Repeated testing of each platform determines whether changes implemented in the training cycle significantly impact operational performance. Each test is carefully constructed to offer maximal training benefits through the use of extensive instructional feedback. Fleet response to the EWCAP for both training and testing has been overwhelmingly positive.

INTRODUCTION

The Electronic Warfare Continuum Assessment Program (EWCAP) is designed to address the issues of electronic warfare (EW) training availability and operational readiness throughout the entire career of Naval aviation personnel. The objectives of the Naval Training Systems Center (NAVTRASYSCEN) EWCAP program are to: (1) determine the EW readiness of the fleet, (2) develop a method for obtaining rapid evaluations of each aviation platform, (3) provide the Chief of Naval Operations (CNO) with documented evidence of training deficiencies, and (4) recommend solutions to problems identified. Sponsors of the EWCAP are CNO, OP-59, and Naval Air Systems Command, PMA-205. With current capabilities, the EWCAP is one of the first concerted efforts to derive training requirements and validate training programs using efficient, automated approaches and empirically-based performance data.

It was required that the program be low cost, provide a quick turnaround, and impose no paper work on the fleet. In order to accomplish the program objectives within this framework, a computer based testing tool [1] was developed under government contract by SWL, Inc., and extensively modified by NAVTRASYSCEN. The Skill and Knowledge Assessment Tool (SKAT) consists of software programs designed to develop and administer tests, and collect data to document levels of knowledge over a wide range of subject matter areas. This software package incorporates both graphics and text to provide the opportunity for complex scenario development and extensive instructional feedback.

EWCAP TEST METHODOLOGY/COMPOSITION

Each test is composed of instructional screens, demographic collection screens, and test question sequences (Figure 1). Each test sequence is composed of the question, remedial instruction or positive feedback, at the minimum. Optionally, a test sequence may also include introductory text or graphics, and a hint screen (Figure 2).

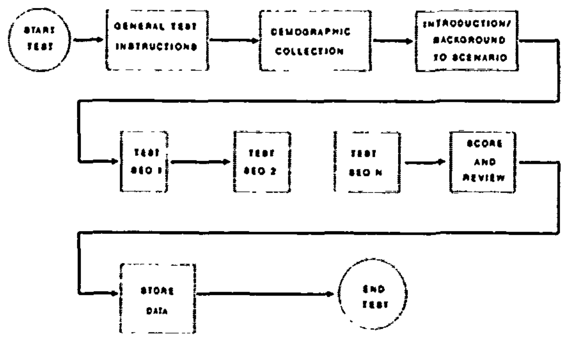


Figure 1: EWCAP Overall Test Sequence

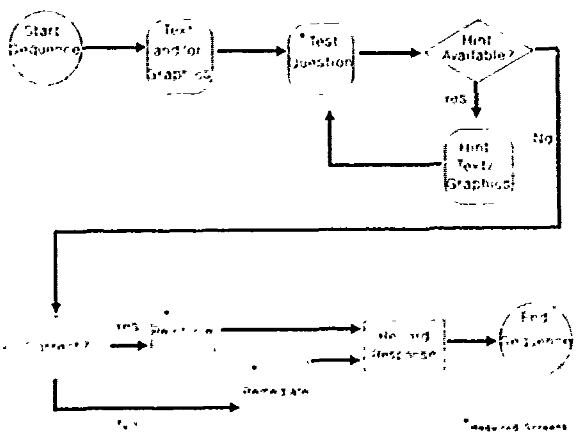


Figure 2: EWCAP Test Question Sequence

Text screens provide the user with tactical background, scenario building information, and battle updates throughout the course of the test. Test screens require a response from the user and are directly followed by remedial or feedback screens. These screens ensure that errors are immediately remediated and misconceptions are not carried through the test.

While the primary goal of the EWCAP is to evaluate the skills and knowledge of the aviation community, each test is carefully constructed to offer maximal training benefits through the use of extensive instructional feedback. Therefore, in addition to correcting errors and clarifying ambiguities, supplemental information related to the question, such as changes in weapons systems, new threat data, or new methods of employing specific tactics can be included. So, in addition to more routine testing and training functions, EWCAP can be used to rapidly disseminate new EW data.

At the conclusion of the test, the user is allowed to review incorrectly answered questions and associated feedback, in order to further resolve misconceptions. The user is given a final score and a summary of performance across categories. Finally, the user is allowed ample space to relay comments back to NAVTRASYSCEN on any facet of the EWCAP test. This information has provided NAVTRASYSCEN important feedback which has lead to significant improvement of the program.

SPECIFIC TEST DEVELOPMENT

Platform specific tests are developed through extensive collaboration between Fleet subject matter experts (SMEs), appointed by the Type or Functional Wings, and NAVTRASYSCEN personnel. These SMEs are primarily EW instructors or EW officers. At working group meetings, mission requirements are discussed at length. The distribution of questions across categories is determined by rating the relative importance of each category to overall mission success. In general, questions fall into one of the following categories: theory, threat, equipment/weapons, offensive and defensive tactics. However, question categories can be tailored to a specific community. For example, the EA-6B community placed more emphasis on EW theory than the F/A-18 community; and the S-3 community combined the tactics categories into an overall integrated EW category.

SMEs generate specific questions, scenarios, introductory material, and required graphics. Questions and scenarios are developed using current threat and tactical data. Several scenarios can be created to serve as a framework for many of the questions throughout a test. Scenarios include geographic displays, electronic orders of battles, intelligence reports, external communications, and battle updates. The scenarios provide a framework in which the user must assimilate and apply tactical and intelligence data to the test problems. Using the specific platform's mission as a context for testing allows for a more realistic evaluation of EW knowledge and skills.

Once development is complete, tests are reviewed by the appropriate platform desk at the Naval Strike Warfare Center. This review ensures that test information does not contradict tactical doctrine. As a final quality control measure, each platform test is given to a small number of operational personnel prior to full Fleet administration.

TEST ADMINISTRATION

EWCAP tests are administered using a computer based testing format. Each aviator is issued a single disk with the full test and demographic survey. Test question responses, latencies, and demographic information are stored on that same disk. All disks are collected and returned to NAVTRASYSCEN for subsequent data analysis.

All aircrew in operational, deployed, and fleet replacement squadrons, are tested. By testing only active fleet personnel, the assessment addresses only those officers who must maintain a high level of EW readiness. Therefore, identified deficits reflect problems in our operational community.

DATA ANALYSIS

Each evaluation identifies specific strengths and weaknesses within the community tested and documents areas requiring remedial and training enhancing actions or policies. The original evaluation for each platform serves as a benchmark for future analyses. Repeated testing of each community will determine whether changes implemented in the training cycle significantly impact operational performance. This allows for documentation that performance changes over time are attributable to factors, such as new training, increased training, and/or the effectiveness of specific courses.

Demographic data collected serve as independent variables. These variables include: rank, operational experience, position in cruise cycle, flight hours, simulator time, EW courses, combat experience, experience with specific systems, mission qualifications, etc. Examination of relationships between these variables and test performance identifies EW deficiencies along with factors that positively impact EW performance. For example, specific courses, time on specific trainers, and position in the cruise cycle can be linked to better performance on the EWCAP. Through examination of these data and discussions with fleet personnel, potential training solutions to identified deficiencies are recommended.

RESULTS

To date, the EWCAP has developed a method of rapid evaluation. This includes software tools for test

development, administration and data reduction as well as a methodology for conducting question bank development and review cycles.

Early findings of the EWCAP demonstrated that: (1) EW capabilities can only be evaluated in the context of the mission, (2) areas of skill emphasis vary widely among platforms, (3) test development must include a broad question base distributed across evaluation categories, and (4) acceptable performance in one community may not be adequate for another [2].

The evaluations of four platforms (EA-6B, E-2C, A-7, and F/A-18) have been completed with two in progress (A-6 and S-3) and two planned (F-14 and EP-3). The question bank is currently at approximately 700 questions and 20 scenarios.

Test development, administration and analysis methods have evolved over the course of the program. During the first testing phase of the A-7 and F/A-18, problems were identified and have been addressed, such as the need for more detailed demographic data collection and a broader question base. The most serious problem encountered in the EA-6B evaluation was that aviators were able to take the test, but exit the program prior to inputting demographic information. This limited the discriminatory value of some of the data collected. A software correction solved this problem for subsequent platform evaluations, by requiring demographic collection at the beginning of the test.

The A-7 and F/A-18 analyses were basically a preliminary Beta testing of the program, but did provide important data highlighting the need for additional training with EW gear and on knowledge of the threat. This was particularly true of the junior officers who lacked operational experience with EW the equipment. Finally, EW performance was found to increase with EW training. The following courses had a positive impact on test performance: EW Officer's Course, Weapons Training Officer's Course, and Strike Leader Attack Training Syllabus (SLATS) Course.

The EA-6B evaluation for Electronic Countermeasure Officers (ECMO) showed that EW scores increased with rank and operational experience, and performance was highest during mid-cruise. Specific courses related to improved performance were also identified (Table 1). This documented the positive impact of current EW training programs.

Table 1
EA-6B ECMO Scores by Specific Courses

Course Title (# Attended)	Attended Course Yes	No
TACAIR Course CNEWS (12)	77%	71% *
Pilot Course CNEWS (3)	87%	71% *
EWO Course VAQ-129 (31)	76%	70% *
SLATS at NSWC (22)	79%	70% *
Med Attack Weap School (44)	77%	69% *

*$p < .05$

The E-2C assessment is nearly complete with additional demographic data collection to include trainer specific information (e.g., Device 15F8, Tactics Trainer, Device 2F110, Operational Flight Trainer, Device 2C20B, Cockpit Procedures Trainer), hours with specific EW systems, and time in each qualification. This assessment, administered late 1990/early 1991, will yield particularly interesting data given its overlap with Desert Shield/Desert Storm activities.

CONCLUSIONS

The EWCAP provides a snapshot of aviator skills and knowledge for each platform. It is this picture that allows for the identification of EW strengths and weaknesses across the spectrum of Naval Aviators' careers, "cradle to grave". Thus, the EWCAP evaluations can be used to define and support training requirements and to defend budgetary plans for implementation of training solutions. Repeated testing provides documentation of the effectiveness of specific training interventions (e.g., new or modified courses and part task trainers).

As previously stated, the EWCAP is carefully constructed to offer maximal training benefits through the use of extensive instructional feedback. Therefore, the EWCAP not only serves to document overall platform readiness, but it also provides direct and immediate diagnostic feedback and training to the individual aviator.

Fleet response to the EWCAP for both training and testing has been overwhelmingly positive. Comments received from EA-6B ECMO's have included: "Very good. Would like to see a training program made available in the same format," "Good review. Like to see more of the same. Valuable training tool," and "Very well written. Enjoyable, showed me my weak areas and I will plan my studying accordingly."

In evaluations to date, perhaps the most notable finding is the strength of specific EW courses. These courses are taken primarily by the more senior officers. Increasing access to the critical information taught in these courses, perhaps through computer-based

training, may decrease performance discrepancies between junior and senior officers.

As EWCAP continues, the program will be further refined. The software itself is in the process of being translated from Basic to C to increase program speed and graphics presentation. More detailed demographic collection, such as the use of specific trainers, will provide more in-depth understanding the differences in performance and assuring efficient training operations across the EW continuum.

The EWCAP's value as an improved and low-cost method for defining training requirements and controlling the quality of training programs is being demonstrated for EW. Application of this methodology could easily be expanded to address the needs of civilian, academic, and other military communities.

REFERENCES

[1] SKAT (1988). SKAT: A Skill and Knowledge Assessment Tool. McLean, VA: SWL, Inc.

[2] Moskal, P. J., Bergondy, M. L., Moser, V. J., and Nida, R. J. (1989). Electronic Warfare Continuum Assessment Program for Aviation. (Report No. NAVTRASYSCEN TR 89-021) Orlando, FL: Naval Training Systems Center.

ABOUT THE AUTHORS

Maureen Bergondy is a Research Psychologist in the Human Factors Division of the Naval Training Systems Center (NTSC). She is currently serving as the Project Manager of the Electronic Warfare Continuum Assessment Program for Naval Aviation and is involved in the NTSC Aircrew Coordination Training research project for the F-14 and A-6. She also served as the Principal Investigator in the development of the Trainer for Radar Intercept Officers. Ms. Bergondy did her undergraduate studies at San Diego State University and was a National Science Foundation Predoctoral Fellow at Vanderbilt University in Experimental Psychology. She has 17 publications and presentations in Experimental Psychology.

Katrina Ricci is a Research Psychologist with the Naval Training Systems Center, Orlando, FL. She has a Masters of Science in Industrial Psychology from the University of Central Florida. She is currently serving as Assistant Project Manager for the Electronic Warfare Continuum Assessment Program for Naval Aviation in the Training Technology Development Branch of the Human Factors Division.

THE USER'S ROLE IN SOURCE SELECTION

Lt Col Ray Willcox
SMSgt Gary Lewis
Tactical Air Command

ABSTRACT

The user's major role and contribution are getting to source selection rather than participation in source selection. The role begins by articulating the requirement and continues through developing the statement of work (SOW), system specification, and evaluation plan. The user may also play a major role in providing clarification of requirements if a draft Request for Proposal (RFP) is issued, and may also provide clarification at the pre-bidder conference.

The user's involvement in getting to source selection may have influenced several trends in RFPs which appear to be emerging. First, there seems to be a preference to reduce risk by specifying proven, commercially available technology rather than emerging technology. Second, there is a desire to require equipment demonstration during the RFP evaluation process. Similarly, the record of past performance by bidding contractors appears to be gaining emphasis in the evaluation process. Finally, the reality of Contracted Logistics Support casts the user as the on-site contract monitors. Consequently, the various aspects of life-cycle support such as spares, support equipment, tech data, and quality assurance plans are of greater interest to the user in developing the RFP.

The user's involvement during the actual source selection is influenced by the perception that there are two fundamental requirements that all training devices must meet. First, they must be concurrent. That is, they must be delivered at the same time, and in the same configuration, as the system they support. Additionally, concurrency means that the training devices can be modified to continue to support training as the weapons system changes or evolves. Second, the devices must be available when needed. Bidders who understand these baseline requirements and are sensitive to emerging trends should be in a strong position to respond to future RFPs.

INTRODUCTION

This paper will concentrate on the user's involvement in getting to source selection. We will identify who the user is, what he wants, and discuss some of the events that users participate in prior to source selection. Additionally, we will provide perspectives on emerging trends in the Request for Proposal process.

WHO IS THE USER?

Identifying the "real" user is as much perspective as it is definition. Industry tends to view its customer--AF Systems Command and AF Logistics Command in the case of Aircrew Training Devices (ATDs)--as the user. The acquisition and support communities view their counterparts at the Operating Commands (TAC, SAC, MAC, ATC) as the user. However, at MAJCOM Headquarters, the user is considered to be the ultimate consumer of the goods or service: the troops at the unit level. This matter of perspective can become further clouded by the fact that HQ TAC acts as the executive agent for the Tactical Air Forces--which includes USAFE, PACAF, and in certain cases, the Air National Guard and AF Reserves. Nevertheless, for the purposes of this paper, the user will be defined as the Operating Command (i.e., Tactical Air Command) and will include all elements from the unit level to MAJCOM Headquarters. One other point for clarification should be mentioned: the common ground of the authors' backgrounds is in the acquisition and utilization of Aircrew Training Devices (ATDs). While a concerted attempt has been made to take a broad perspective towards the user's role in source selection, this ATD context should be kept in mind.

WHAT DOES THE USER WANT?

There are three user requirements that are so fundamental that they are axiomatic. First is concurrency. We need training devices delivered no later than when the weapons system is delivered. When the ATDs arrive, they must be in the same configuration as the weapons system--things work the same way in the training device as they do in the weapons system. Still another aspect of concurrency is that the ATDs can be updated as the weapons system is updated. A brief digression is warranted to acknowledge that concurrency is a tough nut to crack--but it can be done. It requires a genuine team effort between the user, the acquisition community, the prime system contractor, and the training

system contractor. The user needs to state realistic requirements for first article delivered, commonly referred to as the prototype. That is not what the device will ultimately look like: it is the minimum acceptable to support training. Acquisition agencies and the contractor aggressively have to go after those requirements, and that often involves obtaining the support of other System Program Offices (SPOs) (usually the aircraft program office) and other contractors. Adequate testing must be done, but it won't be pure or clean. An ATD can and will be delivered with discrepancies which do not degrade the minimum training requirement. The aircrews need to be involved from day one of testing. Good enough is the goal rather than perfect, but there needs to be sufficient documentation for follow-on fixes and clean up. The acquisition community has to do world-class financial work to keep a good faith contractor adequately funded while still minimizing risk to the Government for the final product. The final two ingredients are honesty and trust. If anyone on the team starts gaming it, the whole deal starts to unravel.

The second fundamental requirement is availability. The training device needs to be up and running when it is scheduled. A device that is dead when it is scheduled for use, especially in formal training courses, or a device that rolls over half-way through a training period, is worse than no device at all. An unreliable training device causes havoc with schedules, and we pay an additional penalty in training with "aircrew acceptance." That's where a particular ATD gets a reputation for being a piece of junk and, by virtue of this attitude, students don't get the full training benefit from the ATD even when it is up and running. (NOTE: Lack of concurrency also breeds this attitude.) Of course, the contractor may eventually pay for this reputation because past performance is a criterion for future source selections.

A third requirement, although not a burning issue at the unit level, is affordability. The present, and foreseeable, fiscal climate is going to shake a lot of trees. Business is not going to be "as usual." Competition is real and it's here to stay, and that competition is going to center on life cycle costs. A "cheap" device that costs a mint to maintain and modify isn't going to remain in the Air Force inventory. Conversely, a high dollar device is no guarantee for longevity. TAC has recently demonstrated a willingness to cut its losses and go to acceptable, rather than optimum, training solutions based on economics.

In summary, the user's role in source selection is influenced by his desire for concurrency, availability, and affordability of the proposal under consideration.

THE USER'S ROLE IN GETTING TO SOURCE SELECTION

The requirement process is a major topic unto itself, and well beyond the scope of this paper. The user plays a central role in this process: need identification, preparing a Statement of Need (SON), and participation through the Program Objective Memorandum (POM) process to ultimately produce a Program Management Directive (PMD) for approved and funded programs. Once the acquisition community is turned on with the PMD, the task of getting to source selection begins in earnest. Note that the user has to assist the acquisition agency to understand the hard-core (not gold plate) requirement so that realistic cost estimates can be prepared. An overpriced program will never get off the ground. The principal document which assists in scoping specifications for the Request for Proposals (RFPs) is the System Operational Requirements Document (SORD). The SORD amplifies and refines the SON and explains how the proposed system will be operated, deployed, employed, and supported. MAJCOM requirements personnel (TAC/DR) build this document. Once the acquisition community has the SORD--which will be updated at each major program milestone--they have all the user inputs necessary, technically speaking, to produce the RFP. However, in a practical sense, the user continues to play an active role.

The SPO may request user participation in developing the Statement of Work, System Specifications, and Evaluation Plan. The extent of participation can range from assistance in writing the first draft to review and coordination on the RFP prior to release. This participation is a mixed blessing for the acquisition folks. On the one hand, user involvement enhances the likelihood that the RFP will really ask for what was intended. On the other hand, it's an ongoing education process within the using command that the official spokesman for requirements is TAC/DR. Every member of the "user family" thinks his vote counts and it does. But they are just votes until DR puts out the official return. Additionally, there are the real-life limitations of providing consistent participation. This can be overcome with a well-documented audit trail. Nevertheless, users are frequently accused of making late changes to the requirements and impeding the process--a

charge not totally lacking validity. In the process of getting the RFP on the street, there may be two opportunities that are to everyone's mutual benefit. The first is issuing a draft RFP to industry for review and comment. In addition to providing potential bidders the greatest amount of information with the most lead time, it also provides a good avenue for industry experts to seek clarification, point out shortcomings or identify potential problem areas. Since it is kosher for all parties to talk with one another at this stage of the game, it is not uncommon for industry to directly question the user. A word of caution: unless the answers come from the SPO, they are not the official answers.

The second opportunity arises at the pre-bidders conference where contractors can interface directly with the ultimate user. This is usually held at either a site where the training device will be delivered and used or at the SPO that is procuring the product. It is a good opportunity for bidders to see where they may be working and to clarify government support questions. It also permits bidders to see where and how their product will be used in the operational environment--the system their product supports. As with questions on a draft RFP, the only official answers come from the SPO.

EMERGING TRENDS IN RFPs

Given the issues that are dear to the user's heart--concurrency, availability, and affordability--and his role in developing the RFP, certain trends seem to be emerging. These trends should be viewed with a "for what it's worth" eye since they are essentially the authors' opinions. Nevertheless, we think they are worth mentioning since there are potential impacts to industry.

First, there seems to be a reluctance to embrace emerging technology to solve training problems and a preference for proven, state-of-the-art, commercially available technology. The reasons for this are fairly obvious. Proven, in production, technology is low risk from both schedule and cost considerations. Documentation is in better shape, support arrangements are more mature, and the R&D costs are already sunk. We may also be a little gun-shy since we took some severe licks betting-on-the-come with emerging technology that never came to fruition or became a financial black hole.

The second trend is related to the first. That is to make an equipment or product demonstration part of the source selection process. Users love this--we understand what we can see. However, it does take a fairly clever SPO to make it work right. Demos have to be conducted against good criteria in order to overcome the natural tendency to rate one product against another. Considerable time should be spent weeding out as much subjectiveness as possible from demonstration objectives. With most off-the-shelf systems, it is possible to have a purely objective criteria as the basis of your up-front demos. Additionally, a good Government team would also have an engineer, a logistician, a program manager, and a knowledgeable user along to make sure there was substance behind the showmanship.

The first two trends give rise to the third. That is a movement, albeit slow, to write performance requirements rather than engineering requirements. Historically, we have taken a requirement for a better mousetrap and written the system specification to include the MILSPEC spring tension, quality and dimensions of the wood, and the FDA standards for the cheese. That gives us firm ground on which to do acceptance testing--which is a wonderful thing. What we really need to do is tell industry that we need something to catch eight mice per day. Then we let the professionals do what they are good at: propose innovative solutions. This isn't as clear as an engineering specification. It assumes that the user really knows what he wants to do with the training device--an assumption not entirely warranted in every case. It also means the traditional testing process (developmental, acceptance, initial operational, and follow-on operational) would require some adjustment. Nevertheless, the potential for schedule and cost savings, as well as the enhancement to competition, would indicate this trend is in everyone's best interests.

The final emerging trend is related to the reality of Contracted Logistics Support (CLS). It has become commonplace for two contracts to be awarded from a single source selection: one for the product; and, one for the contracted support for that product. Consequently, the user has been given on-site contract monitoring responsibility for the performance of CLS. Therefore, we have a greater interest in the various aspects of life cycle support--things like spares, support equipment, tech data, and quality assurance plans--things the user paid little attention to a few years ago.

None of these trends in RFPs is particularly earth-shattering. But the user will arrive at source selection having tried to influence those trends

and armed with the fundamentals that are important to him. The authors would also offer several observations that may be useful to help industry better understand the product user. Observations, of course, are nothing more than opinions that have been dressed up to sound presentable. Therefore, we offer them for "what it's worth."

The first observation deals with the widely recognized move to go to industry for support that has traditionally been performed in-house. There are several examples of this within TAC: Contracted Logistics Support, Contracted Academic Training (CAT), and contracted courseware development and maintenance. It would seem companies who are positioned to provide all those services would be in a competitive position in the future since they can consolidate overhead expenses.

A second observation is that there is increasing interest in buying training, rather than training equipment. For example, a contractor who has simulators used for engineering development may have an asset that would meet a user's training requirement. Time leased on that equipment would allow the contractor to recover some of his capital investment and could be more affordable for the user, compared to the Government trying to buy similar equipment.

The final observation is that there is increased willingness for the user to enter the acquisition arena. This is true in the area of small, command unique programs that are funded from within the MAJCOM. These efforts also tend to be for service (i.e., the CAT program) rather than hardware acquisitions in view of the "kind" of money that we can spend, which is usually O&M 3400 funds.

THE USER'S ROLE IN SOURCE SELECTION

In view of the preceding paragraphs, the user's role in source selection is spectacularly mundane. The source selection authority (AFSC or AFLC) is under no obligation to include the user. However, as a practical matter, they have more work than they can handle, so user participation helps with the load. It also gives the source selection authority an on-scene point of contact to work requirement clarification. Additionally, there may be the tacit, albeit erroneous, assumption that the user is less likely to squawk if he is an accomplice.

Source selection boards are usually divided into four panels: management, technical, logistics, and costs. Users serve on every panel except cost. Budget people have a strong union. Since past performance may be one of the criteria in the management, logistics, and technical evaluations, this is where the "piece of junk" syndrome may have come to roost. Companies that have performed well get good marks. Companies that have a spotted performance may want to consider stepping up to mistakes and detailing how they will avoid them in this new effort.

As users serve on the three panels, working for the Source Selection Authority, they are indistinguishable from the other members except for the tendency to be a little louder and more opinionated. But as they evaluate proposals, they like what they can understand. In other words, substance counts. The user knows that he is going to have to live with the result of the source selection. Therefore, he wants to see and understand how the technical, management, and logistics evaluations are going to meet the need for concurrency and availability.

Two final thoughts on the user's role in source selection. First, the troops who serve on the panels give it a good honest appraisal. Proposals are evaluated against established criteria, not against each other. The sensitivity of the information within the proposal is protected. There is a genuine commitment to giving every bidder a fair shake and an appreciation that preparing a proposal has taken a lot of time and effort. This commitment to honest impartiality is driven by qualities of professionalism and integrity, as well as a healthy respect for the protest process.

Second, users probably get more excited over a protest than does the acquisition community. There is a concern that a protest will impact the program schedule, which impacts training and bringing a weapons system on line. There is the very human reaction to having a good honest appraisal called into question. None of this is said to discourage legitimate protest. Rather, it is intended to restate the obvious: that a company who spends more effort on legal briefs than it does on the proposal is not likely to have a bright business future with TAC programs.

SUMMARY

The user exerts more influence in getting to source selection than he does in the actual process of source selection. Vendors who know their customer and understand his needs should be able to compete and thrive in the

increasingly competitive training industry.

THE AUTHORS

Lt Col Ray Willcox is currently the Director, Air Superiority Simulators, USAF Tactical Air Warfare Center. He was commissioned in 1970 through the Airman's Education and Commissioning Program with a BS from St. Louis University. He also has a MS in Management from Troy State University. Assignments include Strategic Air Command (KC-135s), Air Staff Training Program (ASTRA), F-111s (USAFE), AF Operational Test and Evaluation Center (AFOTEC), and HQ TAC Training Systems Division. He is married to the former Paula Jane Werblin, and has two sons: David (19) and Dan (16). He can be reached at USAFTAWC/TNA, Eglin AFB FL 32542-6008, DSN 872-3327, (904) 882-3327.

SMSgt Gary Lewis is currently the liaison between Tactical Air Command (TAC) and Aeronautical System Division (ASD) for Aircrew Training Devices (ATDs). He enlisted in the Air Force in 1972 and received training as a Navigation/Tactics Training Devices Specialist and has in excess of 18 years of Air Force ATD experience. He received his BS from St Leo College and is currently working towards a MS in Management with Central Michigan University. Assignments included Tactical Air Commands Superintendent of Aircrew Training Devices, HQ TAC, Directorate of Requirements; B-52 Weapons Systems Trainer; F-15 Operational Flight Trainer; F-4D Operational Flight Trainer; and a tour as an Instructor at Chanute Technical Training Center teaching Simulator Operations and Maintenance. He can be reached at OLAO 4525 CAS, Wright Patterson AFB OH 45433-5000, DSN 785-7488, (513) 285-7488.

TACTICAL MISSION TRAINING,

DESIGNING THE VISUAL SYSTEM TO PILOT PERCEPTUAL REQUIREMENTS

Richard J. Heintzman, Heintzman Associates
James E. Brown, Aeronautical Systems Division, USAF
Rick B. Jones, JWK International, Dayton, Ohio

ABSTRACT

To determine visual requirements for ground-based tactical trainers, it is necessary that system designers understand how the aircrew perceives the real world in a tactical situation including what and how various cues are used to accomplish the mission.

Visual simulation system performance requirements are often based purely on visual perceptual data collected under laboratory conditions Such data tends to overstate the requirement since it has no real world or training need modulation and does not reflect the effects of the aircraft and mission environment on human performance. It also does not address factors such as target obscuration or occulting nor how supplementary pointer cues and avionics may be used to locate a target. Data is also needed as to where a pilot looks within the field of view during each element of a mission in order to define field of view requirements of the display.

It is important that the system designer understands the missions and likely conditions and environment that affect the pilot in the real world so that the simulation can reflect these conditions. He must also understand the cues used by the pilot to detect targets, waypoints, SAMS, etc., in order that the data base reflects the proper conditions and supporting cues.

This paper briefly addresses the visual trade process, vision requirements, and the process of collecting and applying pilot perception data to support visual simulation requirements for tactical training in a USAFE type of environment.

INTRODUCTION

Background

United States Air Force Europe (USAFE) together with other members of NATO have been under increasing pressure to reduce low altitude training flights in Germany and other Western European countries because of the potential hazard of such flights to the local population.

With the coming down of the Berlin Wall and the reduction in the Eastern threat, the pressure to restrict low altitude flights geographically to smaller areas and higher altitudes reduces USAFE's ability to train in order to maintain readiness. The European topography makes low altitude terrain following a key tactic for avoiding enemy threats during a USAFE mission such as air interdiction. Training for such missions is a key area of USAFE's defense strategy.

At the request of USAFE, the Training Systems Program Office (SPO) at Aeronautical Systems Division (ASD) contracted with JWK of Annandale, VA. to conduct an indepth Training Systems Requirements Systems Analysis (TSRA). The purpose of the analysis was to define changes and alternatives to the current low altitude mission training program which will allow USAFE to maintain readiness under the flight restrictions that are being imposed. The TSRA was completed in May of 1991. It covered all aspects of both airborne and ground-based training. A key element of the TSRA was a Technology Assessment (TA). Although the TA included a review of technology to support all aspects of ground-based training such as computer based training, procedural training, and part-task training, the most critical area was to assess the ability of visual simulation to support critical aspects of low altitude mission training.

Statement of Problem

The visual system is the most important element of a fighter simulation. The visual system is made up of two principle subsystems, the visual display and the image generator. Low altitude flight and air to surface weapon delivery are the most difficult flight envelopes to simulate. Simulation of air-to-air combat is much simpler because it involves specific high resolution airborne targets and limited ground or surface simulation. Tactical air-to-surface on the other hand, requires detailed terrain simulation over a large gaming area for both low altitude flight to a target area and to perform the attack on a target It includes the simulation of other aircraft, SAM missiles, etc. It also involves flying behind or below terrain features that afford masking from enemy radar and threat forces Low altitude flight subjects the pilot to high work load including a great deal of image scanning. Selecting a display and an image generator which will meet training requirements and at the same time is affordable is extremely difficult. The current high end image

generators under development may meet a good share of the training needs, however their affordability may be questionable Selecting a display system is even more challenging.

To resolve the problem of providing an affordable solution to tactical visual simulation, pilot perception must be a key consideration in the parametric analysis and trade process.

Purpose of the Paper

The purpose of this paper is to describe for the user and the systems designer some of the representative considerations to choose a visual system for tactical simulation and a process to gathering mission related pilot perceptual information. The paper shows how such data may be applied to the visual system selection process with factors such as theoretical perceptual and training effectiveness data.

Approach

The approach taken to conduct the assessment to support mission training was unique. A systems engineering approach was adapted where system performance requirements were first defined from: (1) mission training tasks, (2) human perception requirements and (3) known training effectiveness data. Each major subsystem of a training simulator was first addressed separately and then as part of the trainer system. Data collection included visits to: (1) existing tactical training facilities to gather data on system utility, (2) government R & D facilities to review trainer and training research programs and (3) simulator development contractors to review latest state-of-art in simulation technology. Finally, operational tactical fighter pilots were interviewed to tie together theoretical perceptual requirements with real world pilot experience.

METHOD/PROCESS

Visual Trade Analysis Process

Visual system performance requirements must be based upon low altitude mission training requirements. Therefore, the analysis and recommended configurations were made based upon technology which would support a general set of mission training requirements, training tasks and a base line suite of media. Training effectiveness and perceptual requirements were defined to be part of the data gathering and analysis process. Specific mission training requirements or descriptions such as battle air interdiction (BAI), tactical reconnaissance, and close air support were used as a basis to form the training system performance requirements. These performance requirements together with human perception considerations and training effectiveness considerations provided the basis to define training device subsystem. Later, subsystem trades were conducted to define the most training/cost effective training systems Human perception and training effectiveness data formed a part of the trade process. This was a highly iterative process involving state-of-the-art technology, various trade parameters, together with mission requirements and perceptual and training effective information.

Initially, the human perception data used was theoretical laboratory data. Later pilot perceptual data was incorporated in the trade process in order to modulate the result to account for a real world pilot perception. The pilot inputs turned out to be a key factor to provide a system which includes the proper trades and compromises. The paragraph which follows on human vision and pilot perception discuss some of the more important perceptual factors to be considered for tactical mission training.

Human Vision and Visual Simulation

The human eye is an extremely sensitive high resolution sensing device that even in todays world of exotic high performance electro-optical sensing devices is highly impressive. It is capable of operating under extremely wide ranges of light levels from starlight to a bright day on a beach. It can resolve extremely small details over a wide field-of-view and range of distances. Visual simulation can not duplicate the resolving power or the range of brightness and contrast within which the eye can operate. For that reason we must provide a system which stimulates the eye as similar ranges and level of difficulty that a pilot is expected to experience in a real world tactical situation. However, the limitation in system resolution and brightness will preclude the simulation of conditions such as a clear bright day with very high brightness and contrast. It also means it may not be feasible to train target identification and recognition with a complete range of conditions. However, simulation of enough conditions for visual mission training should be possible.

To develop visual simulation requirements including trading off system performance requirements for an optimum training system, one must first understand the anatomy of the human eye, its performance, and how it perceives information under different conditions.

Eye Characteristics

Several human eye performance characteristics were given careful consideration to develop visual system requirements. Some of the most important include.

(1) Resolution

The central portion of the human eye or foveal region has very high resolution. This area extends roughly only 1.5 to 2 degrees. Human vision beyond the central foveal region is extremely poor.

(2) Field of View

The field of view of human vision with both eyes extends to approximately 200 degrees. Perceptual data indicates that perception of speed and altitude are greatly enhanced by peripheral information. This conclusion was born out by the pilot interviews. Perception of motion is also highly affected by peripheral vision.

(3) Brightness Sensitivity

The human eye is highly sensitive over a wide range of brightness levels. However, very low brightness levels can reduce the resolution of the eye. This should be a concern in the visual system design.

(4) Eye Dynamics

The movement of the eyeball is very important in scanning for and tracking targets and other objects of interest. For certain conditions the eyeball may move as fast as 1,000 degrees per second. Head and body movements may also occur at speeds as high 500 degrees per second. These dynamics must be given careful consideration in the design of the visual display system.

Perception Factors

Although the pilot's eye may perceive detail far better than the visual simulation devices are able to provide such detail, conditions seldom exist in the aircraft which make it possible to achieve such perception. The following factors affect pilots perception during a tactical mission:

(1) Contrast

Contrast is the ability to perceive a lightness or brightness difference between two areas. Generally missions in the real world involve relatively low contrast scenes. Contrast may be enhanced in the simulation to compensate for lower resolution.

(2) Atmospheric Conditions

Atmospheric conditions usually limit the distance at which a target or other object can be perceived in the real world. In a European environment both target contrast and atmospheric clarity will tend to be low. Coupling of low contrast and poor atmospheric conditions can have a significant effect on the pilot's perception of targets and other detail.

(3) Object Occulting

In addition to the effects of low contrast and atmospheric attenuation, a pilot must deal with the obscuration of a target caused by it being occulted by objects in the foreground. This is especially true when dealing with the rolling hilly environment with large amounts of tree cover found in Europe. Flying at low altitudes causes this problem to be extremely severe. Often a pilot may not see a target until such time as he pops-up to perform his attack on a target. Occulting may cause pilots to rely heavily on avionics systems such as radar and FLIR to locate a target.

The factors just discussed together with the human eye characteristics were all taken into account during the analysis of the visual system requirements. Later, their effects on system design were modified and expanded to reflect the inputs received from the pilot interviews.

Rationale and Process Used to Collect Data from Pilots

To determine potential applications, utility, and system requirements of ground-based tactical trainers, it is necessary that system designers understand how the aircrew perceives the real world in a tactical situation. This must include what and how various cues are used to accomplish the mission. It is important that the system designer understands the likely conditions and environment that can affect the pilot in the real world so that the simulation can reflect these conditions i.e., visibility, clouds, overcast, etc. He must also understand the cues used by the pilot to detect targets, waypoints, SAMS, etc., so that the data base properly reflects the proper conditions and supporting cues. Some of the more important factors explored with the pilots in the interview process included (1) the role of peripheral vision, (2) cues used to maintain altitude over terrain, (3) means of tracking over a ridge, (4) effects of weather on performance of a mission, and (5) the role of avionics in a visual mission.

Interview Questionnaire

A questionnaire was prepared which provided structured questioning of the pilots relative to visual cues used in a typical tactical mission in USAFE such as an air interdiction mission. The questionnaire was used as a guide for the interview process. Deviations were made as appropriate during the interviews to assure adequate coverage of issues. Typical of the questions used for the interview are as follows:

(1) How would you fly a wartime air interdiction mission in a high threat environment under different weather conditions in USAFE? (Specific weather conditions were given).

(2) At what range would you expect to be able to detect and recognize different types of targets in USAFE while flying in a high threat environment at an altitude of 300 feet? (Specific targets were given).

(3) How would you judge a ridge crossing in terrain similar to southern Germany?

(4) How much time and how often is the pilot's head in the cockpit while flying low altitude?

(5) How important is peripheral vision and the ability to see the 3-9 line (3 o'clock, 9 o'clock) during a low altitude mission?

(6) What are the cues used to maintain a tactical formation?

(7) What visual cues are used to judge and maintain altitude during low altitude flight? (At 300 feet?, At 500 feet?).

(8) What are the visual responsibilities of the F-15E WSO (weapons systems officer)?

(9) How are air-to-air aircraft targets visually detected and recognized?

Interview Procedure

The questionnaire was validated by interviewing F-16 Air Force reserve pilots from Wright Patterson AFB, Ohio, who had experience flying low altitude in USAFE. Interviews were then conducted with operational F-15E pilots at Seymour Johnson AFB, N. Carolina, and F-16C pilots at Moody AFB, Georgia. There were 20 pilots interviewed at Seymour Johnson AFB and 18 pilots interviewed at Moody AFB. Interview sessions included from 1 to 3 pilots. Interview lengths varied from 45 minutes to 1 hour and 15 minutes. The interviews were recorded in order to insure maximum objective data was collected. Pilots were assured that only background experience level information would be kept and that names would not be kept for record. Recordings appeared to have no affect on pilot responses. The pilots were asked to respond based upon what they would do or possibly experience in a wartime mission. They were asked to ignore peacetime rules of engagement which would probably not exist in a wartime situation.

INTERVIEW RESULTS

Although there was a wide variation in the different points made by the different pilots, there was a high degree of correlation in the responses with very few contradictions. There were some points made by almost every pilot. This high degree of correlation may be due to similar training experiences both on the ground and in the air. The value of an interview did not differ greatly with the degree of experience of the pilot. Experienced pilots made points which the inexperienced pilot did not have the background or exposure to contribute. However, inexperienced pilots often provided better descriptions of certain conditions. This may have been because certain situations had become second nature to the experienced pilot. Also, it was apparent that pilots tend to use different cues as they reach a higher level of experience. As an example, in order to judge altitude, inexperienced pilots will rely more on three dimensional objects to judge altitude, whereas more experienced pilots will rely more on the ground rush (flow field) perceived in his peripheral vision.

Some of the representative responses to the questions are included below in order to provide a feel for the type of information which can be extracted from this type of an interview process:

(1) Question - How would you fly an air interdiction mission in a high threat environment under weather conditions?

Answer - The worse the weather, the fewer the options especially with respect to the type of delivery Delivery affects the type of ordnance which can be used Delivery, together with ordnance affects the ability to destroy the target. Generally speaking, as the weather gets worse, pilot task loading and task management requirements go up. Also, situational awareness goes down, and in the case of a multi-ship engagement, confusion goes up. Under very poor weather conditions, the avionics play a much larger part in the mission.

(2) Question - At what range would you expect to detect and recognize different types of targets in USAFE while flying in a high threat environment at an altitude of 300 feet?

Answers -

(a) Airfields - All pilots agreed that it would be very difficult to detect and recognize a airfield at this altitude because of large trees.

Without vertical development it is possible to fly over an airfield without recognizing it if one approaches the runway other than being parallel to it. More often than not, it may not be visible until one pops-up. This may be anywhere from 2 to 3 miles with some haze or 4 to 5 miles on a clear day. On-board systems such as the radar and/or INS/GPS can help locate it.

(b) Bridges - A bridge may be a very difficult target to locate unless it is very large. At low altitude, the pilot may be required to depend more on his systems than on his eyes. Mission planning is very important. Large pointer cues such as a river and/or road which crosses the bridge and possibly a tree line may be used to pinpoint its location. The bridge may be hidden in a valley in the trees.

There seemed to be a general consensus that bridges would be visible at a distance of somewhere between 1 and 3 miles while flying at low altitude depending on the amount of vertical development, its contrast with the background and the degree to which it was obscured by surrounding vegetation. As an example, a 100 foot long bridge with little vertical development, may be visible at less than a mile. However, with a 30 foot vertical development, it may be visible 2 to 3 miles. A bridge may be visible on the radar if it is in the line of sight. If a bridge has a metal structure, it would provide a very good radar return If the TD box in the HUD is locked to the bridge, it will help direct attention to the bridge to visually identify it. FLIR could also be useful to identify a bridge, if there is a temperature differential between the bridge and its background.

(c) Moving ground targets (tanks, trucks, etc.) - Individual moving targets such as a tank or truck are extremely difficult to detect or recognize. If the vehicles are intent on not being seen from the air, they could successfully hide under trees in a forest. In this case, the one potential clue remaining would be any tracks they would leave from torn up ground and vegetation next to the forest. With some amount of contrast such as being on a road or in an open field they may be visible 1 to 2 miles. Coordination from a FAC (forward air controller) may be useful to find tanks. Also, such effects as dust kicked up by their tracks, track marks in the ground, tracers, smoke, and flashes may be useful cues for their detection. With FLIR, under the right conditions, it may be possible to detect such vehicles out to several miles.

The pilots consistently talked about pre-planning in order to find their way to a target during a mission. They stressed situational awareness which included using large pointer objects which would lead to smaller pointer objects and finally to the IP and the target. They also mentioned that pointer cues should include objects that are permanent and not rely on objects that may have disappeared such as a tower. If FLIR is used, objects should be chosen which will have a temperature differential from the background at the time of day that the mission is to occur. Consideration also needs to be given to picking up pointers which have large amounts of vertical relief that would be visible at low altitude. Visual perception of an area can be affected by the relative positioning of the aircraft, the object interest, and the sun.

(3) Question - What visual cues are used to judge and maintain altitude during low altitude flight?

Answer - A great deal of information was obtained on the different cues that the pilot uses to maintain altitude. These cues tend to be highly altitude dependent and to some degree, experience dependent. They also vary as a function of the type of terrain which the pilot is flying over. All of the pilots said that they use the flight path marker and/or altimeter to set up and calibrate what they see visually. Experienced pilots tended to rely more on the rush of terrain information in the periphery to maintain low altitude. Inexperienced pilots on the other hand tend to rely more on the size of the trees and other objects on the ground. All pilots expressed concern that the use of trees to judge altitude can be a problem since they vary in size in different areas and this can create a false cue. As an example, one pilot said that he had been calibrated flying over large trees and then found himself flying over small brush. This condition was not recognized until a moose appeared which was about the same height as the trees. Although the rush of information in the periphery may appear to be a constant cue, it appears that as the pilot becomes more comfortable at a particular altitude the apparent speed of the rush seems to decrease. If the pilot has not flown at low altitude for a period of time, the speed of the rush appears to be greater until such time as he becomes more comfortable flying at that altitude. Generally the perception of the terrain in the periphery, is not very apparent much above 300 feet. Some of the pilots felt that ground rush goes up exponentially as one flies below 300 feet.

Pilots consistently stated they observed the terrain at low altitude out to about 60 degrees to each side of center line of the aircraft. There were several things discussed that indicate that flying low altitude in rolling terrain is much more difficult than flying over flat terrain In rolling terrain the pilots must have to keep making a mental picture of what appears right Also, in rolling terrain, the lower you are, the less you have of a real horizon

Flat terrain creates another set of problems Comments were made that it tends to "suck you down." Flying over a desert can be extremely difficult because of the lack of any vertical development or texture information. In some cases it may be difficult to tell the difference between 100 and 300 feet. Shadows from sand dunes may help with this problem. Fresh snow is another area which provides almost no cues and is very difficult to judge altitude. One comment made by an experienced pilot with respect to flying at 500 feet was that it helps to fly at that altitude initially in order to learn task management. They said that at this altitude you have more time to do things and that you may only have to spend one fourth of the time concentrating on looking at the terrain than you would at 300 feet.

Inexperienced wingmen appeared to rely heavily on looking at their lead in order to maintain altitude. This is done by placing the lead on the horizon. More experienced pilots said that depending on the lead for altitude maintenance, when weather obscured their horizon or while flying over rolling terrain would not work. They also stated that even over relatively flat terrain, the wingman could not judge his altitude by looking at the lead at 300 feet whereas he could judge his altitude by looking at the lead at 500 feet. It appears that less experienced pilots have not refined their ability to judge altitude at the lower altitude using peripheral flow field information.

(4) Question - How much time and how often is the pilot's head in the cockpit while flying low altitude?

Answer - All the pilots interviewed including F-15E and F-16C, said they spend less than 10% of their time looking in the cockpit at 300 foot altitude. Almost all the pilots said that at 300 feet they would glance in no more than 1 to 2 seconds. At 500 feet, pilots said that they may glance in from 2 to 3 seconds up to 3 to 5 seconds. They all seemed to agree that at 500 feet the lower task saturation made it easier to look in the cockpit.

(5) Question - How important is peripheral vision and the ability to see the 3-9 line during low altitude flight?

Answer - All agreed that peripheral vision is very important for a tactical mission. They said that they needed to see somewhere beyond the 3-9 line in order to maintain line abreast formation and back to the 6 o'clock position in order to visually detect threats. Most pilots mentioned that they required peripheral vision in order to perceive ground rush and judge altitude. There was also a feeling that peripheral vision played an important part in having a "seat of the pants feeling" similar to being in the aircraft.

(6) Question - What are the cues used to maintain a tactical formation?

Answer - Both F-15E and F-16C pilots provided similar responses. They said that it was important for mutual support to see what the other is doing at all times. The distance that they operated varied from 6 to 12,000 feet depending on the aircraft and the weather conditions. As weather conditions became poorer, the tendency was to move in closer and in some cases drop back to a "wedge formation." The bottom line was that a tactical formation was performed at what could be considered eye limiting resolution.

These comments are only a small portion of the responses obtained from the pilots. This includes both the number of questions responded to and the length and breadth of the responses.

INTERVIEW CONCLUSIONS

Although there were variations made in the different points made by the different pilots, there was a high degree of correlation of the information provided. Pilots with different levels of experience and different backgrounds provided different perspectives of how visual perception plays in the performance of a tactical mission. As an example, a former RECCE pilot provided detailed information on how targets are visually located.

Several of the same factors kept coming up in the discussions which seemed to have a large effect on mission success. Some of the most important were situational awareness, task management, and team work. Some of the external forces discussed which affect the performance of the mission are weather, topography, and the threat environment. Weather reduces situational awareness and increases task loading. It also can affect the flight plan, tactics, and the type and way in which the delivery is performed. Rough and rolling topography can also reduce situational awareness, increase task loading and affect the flight plan.

Although, flying in a very low altitude may improve survivability by helping avoid threats, it reduces situational awareness and makes task management more difficult. It also makes visual navigation and target identification more difficult. Whereas, younger pilots sometimes seem to be able to express the way in which different problems are handled, it appears evident that experience and training made such problems much easier to deal with. It also appears that indepth pre-mission planning which anticipates and prepares the pilot to deal with many of the problems that may be encountered can greatly improve situational awareness once into the mission.

Ground rush in the peripheral is the most dependable cue to maintain altitude. However, all pilots indicated that they used vertical development such as trees, buildings, etc. where it exists.

All pilots believed that peripheral vision was extremely important in flying low altitude. They felt that the ground rush came principally in peripheral vision. Several said that it seemed as though they needed about 120 degrees horizontal vision, or visual field-of-view. They also said that during formation flight, they need to see somewhat aft of the 3-9 line and to be able to check 6 for threat aircraft.

Key concerns for any low altitude training should be to facilitate pilot survival ("avoiding the rocks") and accomplishment of the mission.

Impact on Visual System

The conclusions reached from the interviews appear to track well with the perceptual data obtained from various sources. It also appears to mesh well with training effectiveness data which has been generated principally by the Air Force Human Resources Laboratory.

Some of the more important conclusions reached with respect to the implications this data has on visual simulation for tactical mission trainers were as follow:

(1) Avionics including INS/GPS, RADAR, and FLIR must provide the correct real world related cues to the pilot. These systems must closely correlate in content and position with the out-the-canopy visual.

(2) Highly enriched ground information together with three dimensional objects are required to provide the pilot altitude cues for low altitude flight. Also, an instantaneous horizontal field of view of at least 120 degrees together with a full field of regard is required.

(3) Since the pilots periodically make quick glances in the cockpit at low altitude, the visual display should be designed so as not to impair such glances.

(4) A trainer visual system should be designed to provide controlled simulation of illusions. The trainer must also be designed so as not to inadvertently produce illusions which are not related to the real world.

(5) For full simulation of a daylight tactical mission, the instantaneous field of view of the display should at least be 120 degrees and the display should have preferably a full field of regard. A lesser capability display may suffice for a night mission. The visual system must include eye limiting resolution for formation and target aircraft detection and tracking. If necessary this may be done with target projectors.

(6) The WSO has an important visual role in the F-15E. He must have visual display information of the formation aircraft, threats and the terrain. This could drive the overall display system design.

GENERAL CONCLUSIONS

Pilot interviews should play a key role in the definition of a visual system design. Past experience in conducting such interviews has often been less than satisfactory. Many analysts have said that it is impossible to extract such information from pilots. Our experience on this effort was better than we had originally anticipated. Pilots were extremely cooperative, perceptive and articulate in their responses.

We believe that there are several keys to successfully conduct such an effort. To be properly prepared, it is best to first accomplish a preliminary systems definition using mission and task requirements, theoretical perceptual data, and training effectiveness data as available. Once an initial cut is made to define alternative candidate visual subsystems, the analyst should have a better understanding of what is needed from the pilots and the questionnaire can be prepared. The questionnaire should first be validated with interviewees who are representative of the audience to be interviewed. This provides the

analyst with the opportunity to fine tune the questionnaire prior to the final set of interviews. Recording of the interview is also extremely helpful. This takes the heat off the interviewers to record every last detailed remark. Some analysts have expressed concern that the use of a recorder will restrict the interviewee responses. In our case it appeared to have no real impact on the responses which were received. Although we offered each interviewee the opportunity not to use the recorder or turn it off any time they requested us to do so, none of the interviewees made such a request.

Once the interviews are completed and the data has been reduced, the results can readily be incorporated into the visual system analysis.

ACKNOWLEDGEMENTS

The authors wish to thank the fighter pilots at Seymour Johnson AFB and Moody AFB for their time and patience in providing valuable information on pilot perception.

ABOUT THE AUTHORS

Richard J. Heintzman spent twenty six years with the Aeronautical Systems Division of USAF working in the development of aircrew training devices and training systems. He has held many key positions including, Technical Expert, Training Systems, Chief Training Systems Division, Chief Visual and Electro-optical Branch and Program Manager, Simulator for Air to Air Combat (SAAC).

Mr. Heintzman has authored many papers on aircrew training and devices. He was one of the original authors of the Air Force Simulator Master Plan and the Systems Engineering Plan for Manpower, Personnel, and Training. He has been an advisor to the Air Force scientific Advisory Board, the FAA, and foreign Air Forces.

Mr. Heintzman has received the AIAA DeFlorez Training Award and an Air Force Systems Command Award for Scientific Achievement for his technical and management efforts in tactical combat simulation.

For the past three years he has acted as a technical consultant to JWK International and other companies in the field of aircrew training devices and systems.

James E. Brown is Technical Specialist for the Training Requirements Analysis Branch, Training Systems Division, Aeronautical Systems Division, Wright-Patterson AFB, OH. He holds a B.A. in Psychology from Marshall University and an M.S. in Industrial Psychology from North Carolina State University and post-graduate studies at Arizona State University. He is a graduate of the Air War College. He has provided training systems support to the ATF, C-17, C-141, SOF ATS, Joint Primary Aircraft Training System, and USAFE Low-Altitude Training System Requirements Analysis. Jim has over 32 years experience, 10 years in the aerospace industry and 22 years with the government. His government experience includes AFHRL/OT, Williams AFB, AZ, Tactical Air Warfare Center (USAFTAWC) as technical advisor to the Deputy Chief of Staff for Aircrew Training Devices, and since September 1986 has been with ASD/ENET. Jim has over 40 publications.

Mr. Rick Jones currently is the program manager on the Air Force Primary Aircraft Training System (AFPATS) Training System Requirements Analysis (TSRA).
Mr. Jones retired from the Air Force as a Lieutenant Colonel. He was an Instructor Pilot for Undergraduate Pilot Training (UPT) and a FB-111 pilot with SAC.
Mr. Jones was also the B-2 training system program manager while with the Air Force. Mr. Jones has a B.A. in Psychology from the University of Cincinnati and a Masters in management from the State University of New York. Mr. Jones currently works for JWK International Corp at their Dayton, Ohio Air Force Program Office.